MERLIN: MULTI-VIEW REPRESENTATION LEARNING FOR ROBUST MULTIVARIATE TIME SERIES FORECASTING WITH UNFIXED MISSING RATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate Time Series Forecasting (MTSF) aims to predict the future values of multiple interrelated time series and support decision-making. While deep learning models have attracted much attention in MTSF for their powerful spatial-temporal encoding capabilities, they frequently encounter the challenge of missing data resulting from numerous malfunctioning data collectors in practice. In this case, existing models only rely on sparse observation, making it difficult to fully mine the semantics of MTS, which leads to a decline in their forecasting performance. Furthermore, the unfixed missing rates across different samples in reality pose robustness challenges. To address these issues, we propose Multi-View Representation Learning (Merlin) based on offline knowledge distillation and multi-view contrastive learning, which aims to help existing models achieve semantic alignment between sparse observations with different missing rates and complete observations, and enhance their robustness. On the one hand, we introduce offline knowledge distillation where a teacher model guides a student model in learning how to mine semantics from sparse observations similar to those obtainable from complete observations. On the other hand, we construct positive and negative data pairs using sparse observations with different missing rates. Then, we use multi-view contrastive learning to help the student model align semantics across sparse observations with different missing rates, thereby further enhancing its robustness. In this way, Merlin can fully enhance the robustness of existing forecasting models to MTS with unfixed missing rates and achieves high-precision MTSF with sparse observations. Experiments on four real-world datasets validate our motivation and demonstrate the superiority and practicability of Merlin.

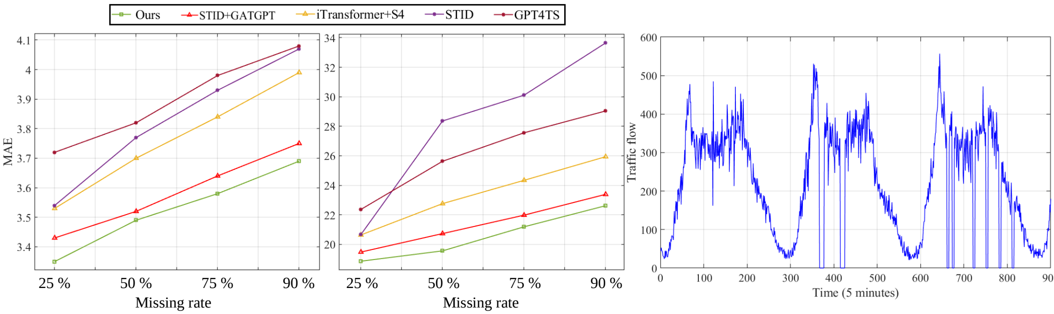
1 INTRODUCTION

Multivariate Time Series Forecasting (MTSF) is widely used in practice, such as transportation (Wang et al. (2023)), environment (Tan et al. (2022)) and weather (Xu et al. (2021)). Deep learning-based models, such as Spatial-Temporal Graph Neural Networks (STGNNs) (Shao et al. (2022b)) and Transformers (Yu et al. (2023b)), are widely used due to their powerful semantic mining capabilities (Benidis et al. (2022)). However, they need to fully mine semantics (Global and local information) from the complete MTS, and achieve accurate spatial-temporal forecasting (Zheng et al. (2020)). In reality, due to factors such as natural disasters and component failures, data collectors can easily malfunction and fail to output data normally (Zheng et al. (2023)). In this case, existing models only use sparse observations to predict future values, which limits their performances (Cini et al. (2022)). To illustrate, we evaluate the performance of several models (Liu et al. (2023); Shao et al. (2022a); Zhou et al. (2023)) under different missing rates on the METR-LA dataset and PEMS04 dataset. As shown in Figure 1(a) and Figure 1(b), the forecasting errors (Mean Absolute Error) of existing models increase continuously as the missing rate increases.

To mitigate the adverse effects of incomplete MTS data, we must delve deeper into a question, that is, **how does MTS data missing fail these models?** By rethinking the characteristics of this task, we believe that a large number of missing values[1] in historical observations can severely disrupt

---

[1] Missing values in most datasets, such as PEMS04 and METR-LA, are usually processed as zeros.



(a) MAE values of different models (METR-LA) **(b) MAE values of different models (PEMS04)** **(c) Traffic flow data with missing values.**

Figure 1: Examples of MTSF with sparse observations. (a) MAE values of different models on METR-LA. (b) MAE values of different models on PEMS04. As the missing rate increases, the forecasting errors of several models continue to rise. (c) Missing values disrupt the global information in time series (such as periodicity), and introduce error local information (such as sudden changes). Furthermore, the missing rate of time series changes over time.

the semantics of MTS and affect the robustness of forecasting models. Specifically, as shown in Figure 1 (b), on the one hand, missing values disrupt the global information (such as periodicity) of time series and introduce error local information such as sudden changes (From normal to zero) and abnormal straight lines. If models forcibly capture these anomalies, they will mine incorrect semantics, leading to a decline in forecasting accuracy. On the other hand, since the distribution of missing values usually changes over time, the missing rates of time series at different time points are often unfixed. In this case, existing models (Lim et al. (2021); Li & Zhu (2021); Tang et al. (2020)) often need to be trained separately for different missing rates to ensure their performance, further limiting their practicability. These two phenomena lead to existing models having poor robustness in MTSF with sparse observations, resulting in a decline in their forecasting performance.

Based on the findings above, we believe that the core reason why existing forecasting models fail to achieve effective forecasting results in MTSF with sparse observations is that missing values inhibit their ability to accurately capture the semantics in sparse observations and limit their robustness. To solve the above problems, existing works use imputation methods to improve the performance of forecasting models and propose two-stage modeling approaches (Xu et al. (2023)) or end-to-end modeling approaches (Tran et al. (2023)) to improve their performance. However, these methods still face several challenges: (1) Existing imputation methods (Miao et al. (2021); Wu et al. (2023a)) usually require reconstructing both missing and normal values, which can disrupt the local information of MTS and lead to error accumulation. (2) Existing imputation methods (Du et al. (2023); Zhou et al. (2023)) need to train models separately for data with different missing rates to ensure the accuracy of data recovery. Since the missing rates in MTS are often unfixed at different time points in reality, existing imputation methods struggle to effectively recover time series with unfixed missing rates, which limits their robustness and practicality. Overall, imputation methods still fail to fully assist forecasting models in accurately mining semantics from sparse observations and addressing the issue of poor robustness. As shown in Appendix H, if imputation and forecasting models are not trained separately for each missing rate, their performance is limited.

To solve the above problems and realize robust multivariate time series forecasting with unfixed missing rates, we need to enhance the capability of existing forecasting models for semantic alignment, which includes two aspects: (1) enabling forecasting models to align the semantics between sparse observations and complete observations. (2) enabling forecasting models to align the semantics among sparse observations with different missing rates. To this end, we propose Multi-View Representation Learning (Merlin) by taking advantage of knowledge distillation and contrastive learning. On the one hand, knowledge distillation can transfer valuable knowledge from the teacher model to the student model, thereby constraining the modeling process of the student model and improving its performance (Dong et al. (2023)). Considering that the model can mine more accurate semantics with complete data, we use the model trained with complete data as the teacher model. The student model, whose input features are sparse observations, has the same structure as the teacher model. In the training process, we transfer representations and forecasting results obtained by the teacher model as knowledge to the student model, aiming to make the student model produce representations and forecasting results that are as similar to them as possible. In this way, by constraining the student model's encoding process and forecasting process, it can learn how to align the semantics between sparse observations and complete observations, thereby enhancing the quality of the semantics mined by the student model. On the other hand, multi-view contrastive learning can help the model enhance the semantic differences in negative data pairs and the semantic similarities in positive data pairs (Zhang et al. (2024)). To further achieve semantic alignment between samples with different missing rates and enhance the robustness of the student model, we treat samples from the same time point with different missing rates as positive data pairs, and samples from different time points as negative data pairs. In this way, multi-view contrastive learning strengthens the ability of the student model to mine and align the semantics of sparse observations with different missing rates. In this way, we only need to train one student model to adapt to unfixed missing rates, significantly enhancing its robustness. Based on above methods, Merlin can effectively help existing forecasting models learn how to mine semantics from sparse observations, just as if using complete observations. Additionally, Merlin can enhance the ability of existing forecasting models to achieve semantic alignment between sparse observations with different missing rates, enabling them to achieve robust multivariate time series forecasting with unfixed missing rates. **The main contributions of this paper can be outlined as follows:**

- We believe that the main issue limiting the performance of existing forecasting models in MTSF with sparse observations is their poor robustness. On the one hand, missing values introduce error semantics (such as sudden changes) to time series. On the other hand, the missing rate of time series often changes over time, and existing models need to be trained separately for different missing rates.
- We believe that the key to achieving robust MTSF with unfixed missing rates is to help existing models achieve semantic alignment between sparse observations with different missing rates and complete observations. To this end, we propose Multi-View Representation Learning (Merlin), including knowledge distillation and contrastive learning.
- We design experiments on four real-world datasets. Results show that Merlin can enhance the performance of existing forecasting models more effectively than other imputation methods. Besides, through Merlin, forecasting models only need to be trained once to adapt to sparse observations with different missing rates.

2 RELATED WORK

2.1 SPATIAL-TEMPORAL FORECASTING METHODS

Classic STGNNs (Liu et al. (2021); Li et al. (2018); Wu et al. (2019)) combine the Graph Convolutional Network (GCN) and sequence models to exploit spatial-temporal correlations. Besides, existing advanced STGNNs (Yi et al. (2023); Yu et al. (2024a)) introduces graph learning technology to further improve the ability of modeling spatial correlations. Different from STGNNs, existing Transformers (Wu et al. (2023b); Zhang & Yan (2022); Yu et al. (2023a)) combine temporal attention and spatial attention, or their variants, to capture spatio-temporal information. Although STGNNs and Transformers have achieved extensive research, they often suffer from high complexity and limited scalability (Yu et al. (2024b)). Currently, lightweight models based on Multi-Layer Perceptron (MLP) have gained widespread recognition. (Chen et al. (2023a)) proposes TSMixer, which use all-MLP architecture to mine spatial-temporal correlations. (Shao et al. (2022a)) analyze the core of modeling spatial-temporal correlations and propose an MLP framework based on the spatial-Temporal Identity (STID). In summary, a suitable MLP framework can achieve satisfactory results more efficiently than complex models. Considering that STID analyzes the characteristics of MTSF and has satisfactory performance on most datasets, it is selected as the backbone. Besides, we also evaluate the performance improvement of Merlin on other complex models.

2.2 KNOWLEDGE DISTILLATION

Knowledge distillation can transfer valuable knowledge from the teacher model to the student model to improve the student model's performance (Xu et al. (2022)). Mainstream techniques include offline knowledge distillation and online knowledge distillation. Among them, offline knowledge distillation offers advantages such as good stability, high flexibility, and a simplified training process. It improves the ability of the student model by continually guiding it to align with the teacher model (Yang et al. (2022)). (Chattha et al. (2022)) use knowledge distillation to enhance the ability of neural networks to mine samples. Experiments show that the proposed method can still achieve satisfactory results even if the sample size is reduced by 50%. (Monti et al. (2022)) propose a trajectory forecasting model based on knowledge distillation and spatial-temporal Transformer, enabling the student model to perform well with only 25% of historical observations. In summary, knowledge distillation can help the student model achieve satisfactory forecasting results even when the effective information in input features is significantly reduced (Wang et al. (2021)). Therefore, it can enhance the student model's capability to handle sparse observations.

2.3 CONTRASTIVE LEARNING

Multi-view contrastive learning enhances the model's ability to mine key information by aligning the semantics of the similar samples under different views (Hassani & Khasahmadi (2020)). (Woo et al. (2021)) treat the seasonal and trend components of time series as different views and use contrastive learning to align the semantics of these different views. (Yue et al. (2022)) propose the hierarchical contrastive learning method to help the model improve their ability to align the semantics of time series with different scales. (Liu & Chen (2023)) propose a self-supervised contrastive learning framework for time series representation learning, and make the forecasting model produce more reliable representations. (Dong et al. (2024)) combine different masking ways with contrastive learning to mine semantics from time series. Experimental results show that contrastive learning aligns the semantics of different masked time series and enhances the reconstruction effect. Based on these references, it can be found that contrastive learning can enhance the model's ability to distinguish different samples and align the semantics between positive data pairs (Liu et al. (2022)). Therefore, if we can effectively construct positive data pairs, contrastive learning can align the semantics of sparse observations with different missing rates and enhance the model's robustness.

3 METHODOLOGY

3.1 PRELIMINARIES

In this section, we introduce multivariate time series forecasting and multivariate time series forecasting with sparse observations. Some of the commonly used notations are presented in Table 1.

Multivariate time series forecasting (Chengqing et al. (2023)). Given a historical observation tensor $X \in R^{N_v * N_H * N_c}$ from N_H time slices in history, the model can predict the value $Y \in R^{N_v * N_L}$ of the nearest N_L time slices in the future. N_v is the number of sequences. N_c is the number of features. The core goal of MTSF is to construct mapping function between input $X \in R^{N_v * N_H * N_c}$ and output $Y \in R^{N_v * N_L}$.

Multivariate time series forecasting with sparse observations (Sridevi et al. (2011)). Compared with MTSF, the main difference of this task is that there are so much missing values in historical observations. In other words, we need to mask $M\%$ point randomly from the historical observation tensor $X \in R^{N_v * N_H * N_c}$. After the above processing, a new input feature $X_M \in R^{N_v * N_H * N_c}$ is obtained. The core goal of this task is to construct mapping function between input $X_M \in R^{N_v * N_H * N_c}$ and output $Y \in R^{N_v * N_L}$.

3.2 OVERALL FRAMEWORK

The overall framework of Merlin is shown in Figure 2. During the training phase, we utilize STID as the backbone and propose Merlin that combines offline knowledge distillation with multi-view contrastive learning to it. At this stage, the input features of the teacher model are complete historical observations. The input features of the student model are sparse observations. During the inference phase, we only use the student model for forecasting, whose input features are sparse observations with different missing rates. Next, we briefly describe the motivation for designing each component.

First, we explain the motivation for using STID as the backbone, which has the following advantages: (1) It introduces spatial-temporal identity embeddings to provide the model with additional

Table 1: Frequently used notation.

Notation	size	Definitions
N_H	Constant	Length of historical observations
N_L	Constant	Length of forecasting results
N_s	Constant	Batch size
N_v	Constant	Number of variables
N_c	Constant	Number of features
m	Constant	Number of missing rates
X	$N_v * N_H * N_c$	Complete historical observations
X_M	$N_v * N_H * N_c$	Sparse observations
Y	$N_v * N_L$	Forecasting results
FC	Functions	Fully connected layer
ReLU	Functions	Activation function ReLU
Mean	Functions	The mean of the Tensor
softmax	Functions	Activation function softmax

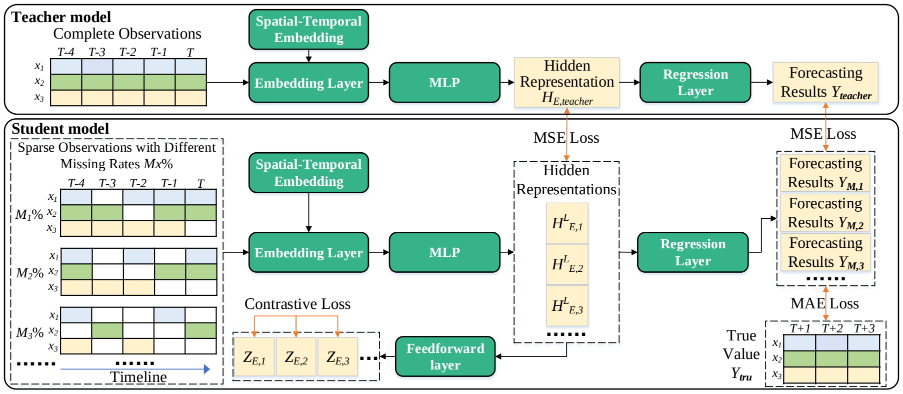


Figure 2: Overall framework of Merlin. During the training phase, the inputs of the teacher model and the student model are complete observations and sparse observations respectively. During the inference phase, only the student model is used for forecasting, whose inputs are sparse observations.

information, effectively mitigating the damage of missing values. (2) It adopts a lightweight framework, which results in the model's computational complexity being only $O(N_H)$.

Then, we briefly introduce offline knowledge distillation, whose purpose is to enable STID to learn how to align the semantics between sparse and complete observations. We first train STID as a teacher model using complete observations. Then, when training the student model using sparse observations, we transfer the knowledge of the teacher model by using the representations and forecasting results generated by the teacher model. This helps the student model learn how to use sparse observations to generate representations and forecasting results similar to those generated by the teacher model. In this way, the student model can achieve semantic alignment between sparse observations and complete observations as much as possible.

Finally, we discuss the effects of multi-view contrastive learning. Although offline knowledge distillation helps STID learn how to align the semantics between sparse and complete observations, the student model still needs to improve its robustness to unfixed missing rates. Therefore, the student model needs to learn how to align the semantics between sparse observations under different missing rates. Therefore, we use sparse observations under different missing rates as positive data pairs and different samples within the same batch as negative data pairs. Through this method, the student model can utilize multi-view contrastive learning to enhance its robustness to sparse observations with different missing rates, without the need for retraining.

3.3 BACKBONE

In this section, we briefly introduce the basic structure of the backbone (STID), which is composed of a embeding layer, L fully connected layer and a regression layer. A detailed description and definition of STID can be found in the reference (Shao et al. (2022a)). The basic modeling process of STID is shown as follows:

Step I: First, the embedded layer based on a fully connected layer is used to transform the input feature X into a high dimension hidden representation H:

$$H = \text{FC}(X), \tag{1}$$

where, FC$(\cdot)$ is the fully connected layer.

Step II: Then, the spatial-temporal identity embedding (S_E, T_E^D and T_E^W) are passed to H as additional inputs to improve the ability of the encoder to produce effective representations.

$$H_E = \text{Concat}(H, S_E, T_E^D, T_E^W), \tag{2}$$

where, Concat$(\cdot)$ means concatenate several tensors. Assuming N_v time series and N_H time slots in a day and N_w = 7 days in a week. $S_E \in R^{N_v * D}$ is the spatial identity embedding. $T_E^D \in R^{N_H * D}$ and $T_E^W \in R^{N_w * D}$ are the temporal embedding. D is the embedding size.

Step III: The encoder based on L layers of MLP with the residual connection is used to mine the above representation Z. The l-th MLP layer can be denoted as:

$$H_E^{l+1} = \text{FC}(\text{Relu}(\text{FC}(H_E^l))) + H_E^l, \tag{3}$$

where, $Relu(\cdot)$ is the activation function.

Step IV: Finally, based on the hidden representation H_E^L, the regression layer is used to obtain the forecasting results Y.

$$Y = \text{FC}(H_E^L), \tag{4}$$

In the following section, we will show how to use the hidden representation H_E^L and forecasting result Y for knowledge distillation and contrastive learning.

3.4 OFFLINE KNOWLEDGE DISTILLATION

In this paper, we use two STID models as the student model and the teacher model. The input features to the teacher model are the complete historical observations X. It produces the hidden representation $H_{E,Teacher}^L$ and the forecasting result $Y_{Teacher}$. The input features to the student model are the sparse observations $X_{M,1}$ to $X_{M,m}$. m stands for the number of missing rates. It produces m hidden representations $H_{E,1}^L$ to $H_{E,m}^L$ and m forecasting results $Y_{M,1}$ to $Y_{M,m}$.

The offline knowledge distillation consists of two components: the hidden representation distillation and the forecasting result distillation. The hidden representation distillation refers to transferring the representations produced by the teacher model to the student model, aiming to minimize the mean squared error (MSE) between the representations produced by the student model and those produced by the teacher model. Its specific formula is shown as follows:

$$L_{HD} = \frac{1}{m}(\sum_{i=1}^{m} \text{Mean}((H_{E,Teacher}^L - H_{E,i}^L)^2)), \tag{5}$$

where, $Mean(\cdot)$ is the mean of the Tensor.

The process of forecasting result distillation involves transferring the forecasting results produced by the teacher model to the student model, with the objective of minimizing the MSE between the forecasting results produced by the student model and those produced by the teacher model. The specific formula is shown as follows:

$$L_{RD} = \frac{1}{m}(\sum_{i=1}^{m} \text{Mean}((Y_{Teacher} - Y_{M,i})^2)), \tag{6}$$

Based on L_{HD} and L_{RD}, the teacher model can effectively guide the student model to use sparse observations to produce better representations and forecasting results. In this way, the student model can effectively achieve semantic alignment between sparse observations and complete observations, thereby enhancing its ability to mine key semantics from sparse observations.

3.5 MULTI-VIEW CONTRASTIVE LEARNING

Considering that the missing rates of historical observations in reality are not fixed, in order to further enhance the robustness of the student model and realize the semantic alignment of data with different missing rates, this paper proposes a multi-view contrastive learning method. We use historical observations with different missing rates at the same time point as positive data pairs, and use historical observations at different time point (other samples within a batch) as negative data pairs. For representations $H_{E,1}^{L}$ to $H_{E,m}^{L}$ encoded by historical observations with different missing rates, we employ a pairwise contrastive learning approach to achieve multi-view contrastive learning. The specific steps are given as follows:

Step I: Considering that appropriate dimension reduction can enhance the effectiveness of contrastive learning, a fully connected layer is used to decode the hidden representations $H_{E,1}^{L}$ to $H_{E,m}^{L}$, and get the representations $Z_{E,1}$ to $Z_{E,m}$ for Contrastive learning.

$$Z_{E,1} = \text{FC}(H_{E,1}^{L}), \tag{7}$$

Step II: Firstly, we use the $Z_{E,1}$ and $Z_{E,2}$ to obtain $2N_s$ samples. In $Z_{E,1}$ and $Z_{E,2}$, the corresponding two samples form a positive data pair, while the other samples are their negative data pairs. The contrast loss between any two samples $z_{E,i}$ and $z_{E,j}$ is shown as follows:

$$l_{i,j} = -\text{log}\left(\frac{\text{exp}(\text{sim}(z_{E,i}, z_{E,j})/\tau)}{\sum_{k=1\&k\neq i}^{2N_s} \text{exp}(\text{sim}(z_{E,i}, z_{E,k})/\tau)}\right), \tag{8}$$

where, $\text{exp}(\cdot)$ is the exp function. $\text{sim}(\cdot)$ is the Cosine similarity. N_s is the number of samples. τ is the temperature parameter.

Step III: Then, the contrastive loss between $Z_{E,1}$ and $Z_{E,2}$ can be obtained by the following formula:

$$L_{Z1,Z2} = \frac{1}{2N_s}\sum_{k=1}^{N_s}(l_{2k-1,2k} + l_{2k,2k-1}), \tag{9}$$

Step IV: Repeat the above steps and obtain the contrastive loss between $Z_{E,1}$ to $Z_{E,m}$ pairwise. The final multi-view contrastive learning loss is shown below:

$$L_{CL} = \frac{2}{m(m-1)}\left(\sum_{Zj=Zi}^{m}\sum_{Zi=1}^{m-1} L_{Zi,Zj}\right), \tag{10}$$

3.6 LOSS FUNCTION

To realize the supervised learning process, we also incorporate ground truth and L1 loss to train the student model. The formula is shown as follows (Challu et al. (2023)):

$$L_{Pre} = \frac{1}{m}\left(\sum_{i=1}^{m} \text{Mean}(|Y_{tru} - Y_{M,i}|)\right), \tag{11}$$

where, Y_{tru} is the ground truth. $|\cdot|$ stands for absolute value.

Finally, we need to effectively combine all the above Loss functions. There are two main ways to integrate these Loss functions (Gou et al. (2023)): multi-stage training or stacking all Loss functions. Considering the problem of information forgetting caused by multi-stage training, we use the method of adding all Loss functions. The formula is given as follows:

$$L_{Finally} = L_{Pre} + \beta(L_{HD} + L_{RD} + L_{CL}), \tag{12}$$

where, β stands for the weight of the Loss. After completing the process of the training phase, the inference phase is performed by using only the student model. Besides, the input features are sparse observations with different missing rates.

4 EXPERIMENT AND ANALYSIS

4.1 EXPERIMENTAL DESIGN

Datasets. To comprehensively evaluate the validity of the proposed model, we select four real-world datasets from different domains: traffic speed (METR-LA), traffic flow (PEMS04), environment (China AQI), and meteorology (Global Wind). Detailed descriptions are provided in Appendix A.1.

Baselines. To comprehensively verify the performance of the proposed model, we select baselines from three perspectives: (1) We select three one-stage models that can handle missing values: GPT4TS (Zhou et al. (2023)), MegaCRN (Jiang et al. (2023)), and Corrformer (Wu et al. (2023b)). (2) To demonstrate the improvement of Merlin on STID, we compare the STID+Merlin with the raw STID. Besides, we select four imputation methods and combine them with STID to create multiple two-stage models: STID+GATGPT (Chen et al. (2023c)), STID+SPIN (Ivan et al. (2022)), STID+GPT2 (Zhou et al. (2023)) and STID+MAE (Li et al. (2023)). (3) We combine several existing spatial-temporal forecasting models with imputation models, and obtain several two-stage models as baselines: iTransformer (Liu et al. (2023)) + S4 (Gu et al. (2022),) FourierGNN (Yi et al. (2023)) + SPIN, DSformer (Yu et al. (2023a)) + GATGPT, and TSMixer (Chen et al. (2023a)) + GPT2 (Note: The previous method for each combination is the forecasting model.).

Setting. Hyperparametric analysis can be found in the Appendix B. Besides, we design the experiments from the following aspects: (1) According to ratios in (Shao et al. (2023)), four datasets are uniformly divided into training sets, validation sets, and testing sets. (2) The history length and future length of all forecasting models are 12. All Metrics are calculated as the average of the 12-step forecasting results. More experiments on the history length and future length can be found in the Appendix G and Appendix B. (3) We randomly assign mask points with ratios of 25%, 50%, 75%, and 90%. The value of the masked point is uniformly set according to related works (Chen et al. (2023b)). Experiments are repeated with 5 different random seeds for each model. The final metrics are calculated as the mean value of repeated experiments. In addition, we provide the standard deviation of the forecasting results. (4) To prove the robustness of our model, we train it once, using samples with multiple missing rates. In other words, the student model is trained simultaneously using data with missing rates of 25%, 50%, 75%, and 90%. For other baselines, we train them using two ways and report the best results: one is training a separate model for each missing rate, and the other is training a single model using samples with multiple missing rates (Shan et al. (2023)).

Metrics. In order to comprehensively evaluate the forecasting performance of our model and other baselines, three classical metrics are used, including MAE (Mean Absolute Error), RMSE (Root Mean Square Error) and MAPE (Mean Absolute Percentage Error) (Liu et al. (2020)).

4.2 MAIN RESULTS

Table 2 shows the performance comparison results of all baselines and the proposed model on all datasets. Based on the experimental results, we can draw the following conclusions: (1) Compared with other two-stage models, the forecasting errors of all single-stage models are larger. The main reason is that existing single-stage models are easily affected by missing values, leading them to mine incorrect semantic. (2) Compared with other imputation methods, Merlin can improve the forecasting performance of STID more effectively. The main reason is that Merlin effectively combines the advantages of multi-view contrastive learning and offline knowledge distillation, which can significantly enhance the robustness of STID in modeling sparse observations and improve the capacity of STID to mine the semantics from data. (3) STID+Merlin can work better than all baselines in all cases. Firstly, we select the high-performance STID as our backbone model, which introduces temporal and spatial embeddings to provide additional semantic information for the model, helping to mitigate the impact of missing values. Secondly, we introduce offline knowledge distillation to instruct STID on how to align the semantics between sparse observations and complete observations, thereby enhancing the model's ability to mine crucial information. Finally, we propose multi-view contrastive learning to achieve semantic alignment among sparse observations with different missing rates, further improving the robustness of STID. Therefore, STID+Merlin can achieve the best forecasting results on all datasets and all missing rates. In the next section, we will further evaluate Merlin's performance improvement effects on other backbone models.

Table 2: Performance comparison results of several models. The best results are shown in **bold**. The subscript represents the standard deviation of the forecasting results.

Datasets	Models	Missing rate 25%			Missing rate 50%			Missing rate 75%			Missing rate 90%		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
METR-LA	Corrformer	3.74$_{\pm0.02}$	10.56$_{\pm0.10}$	7.22$_{\pm0.04}$	3.88$_{\pm0.02}$	11.15$_{\pm0.12}$	7.62$_{\pm0.04}$	3.97$_{\pm0.04}$	11.71$_{\pm0.14}$	7.94$_{\pm0.06}$	4.15$_{\pm0.04}$	12.38$_{\pm0.18}$	8.25$_{\pm0.7}$
	MegaCRN	3.63$_{\pm0.02}$	10.13$_{\pm0.10}$	6.88$_{\pm0.04}$	3.79$_{\pm0.02}$	10.76$_{\pm0.12}$	7.38$_{\pm0.04}$	3.94$_{\pm0.04}$	11.18$_{\pm0.14}$	7.65$_{\pm0.02}$	4.03$_{\pm0.04}$	11.89$_{\pm0.17}$	7.93$_{\pm0.06}$
	GPT4TS	3.72$_{\pm0.02}$	10.49$_{\pm0.10}$	7.21$_{\pm0.04}$	3.82$_{\pm0.02}$	10.86$_{\pm0.11}$	7.39$_{\pm0.04}$	3.98$_{\pm0.04}$	11.31$_{\pm0.14}$	7.75$_{\pm0.06}$	4.08$_{\pm0.04}$	12.01$_{\pm0.15}$	8.04$_{\pm0.07}$
	iTransformer+S4	3.53$_{\pm0.02}$	9.43$_{\pm0.10}$	6.74$_{\pm0.04}$	3.70$_{\pm0.02}$	10.31$_{\pm0.12}$	6.97$_{\pm0.04}$	3.84$_{\pm0.02}$	10.91$_{\pm0.13}$	7.42$_{\pm0.04}$	3.99$_{\pm0.04}$	11.44$_{\pm0.14}$	7.86$_{\pm0.06}$
	FourierGNN+SPIN	3.50$_{\pm0.01}$	9.32$_{\pm0.08}$	6.71$_{\pm0.02}$	3.63$_{\pm0.01}$	10.15$_{\pm0.08}$	6.89$_{\pm0.02}$	3.75$_{\pm0.02}$	10.79$_{\pm0.10}$	7.34$_{\pm0.04}$	3.91$_{\pm0.02}$	11.24$_{\pm0.13}$	7.68$_{\pm0.04}$
	DSformer+GATGPT	3.52$_{\pm0.01}$	9.37$_{\pm0.09}$	6.73$_{\pm0.02}$	3.65$_{\pm0.01}$	10.24$_{\pm0.09}$	6.94$_{\pm0.02}$	3.78$_{\pm0.02}$	10.86$_{\pm0.10}$	7.38$_{\pm0.04}$	3.89$_{\pm0.02}$	11.19$_{\pm0.12}$	7.66$_{\pm0.04}$
	TSMixer+GPT2	3.48$_{\pm0.01}$	9.29$_{\pm0.08}$	6.69$_{\pm0.02}$	3.62$_{\pm0.01}$	9.97$_{\pm0.09}$	6.85$_{\pm0.02}$	3.71$_{\pm0.02}$	10.48$_{\pm0.10}$	7.25$_{\pm0.04}$	3.85$_{\pm0.02}$	11.14$_{\pm0.12}$	7.65$_{\pm0.04}$
	STID (Raw)	3.54$_{\pm0.02}$	9.35$_{\pm0.10}$	6.74$_{\pm0.04}$	3.77$_{\pm0.02}$	10.83$_{\pm0.12}$	7.29$_{\pm0.04}$	3.93$_{\pm0.04}$	11.16$_{\pm0.14}$	7.64$_{\pm0.07}$	4.07$_{\pm0.04}$	11.89$_{\pm0.16}$	8.03$_{\pm0.08}$
	STID+SPIN	3.44$_{\pm0.01}$	9.27$_{\pm0.07}$	6.65$_{\pm0.02}$	3.54$_{\pm0.01}$	9.36$_{\pm0.08}$	6.75$_{\pm0.02}$	3.67$_{\pm0.02}$	10.44$_{\pm0.12}$	7.05$_{\pm0.04}$	3.79$_{\pm0.02}$	10.92$_{\pm0.13}$	7.41$_{\pm0.04}$
	STID+GPT2	3.49$_{\pm0.01}$	9.31$_{\pm0.08}$	6.68$_{\pm0.02}$	3.59$_{\pm0.01}$	9.44$_{\pm0.09}$	6.79$_{\pm0.02}$	3.68$_{\pm0.02}$	10.46$_{\pm0.10}$	7.09$_{\pm0.04}$	3.77$_{\pm0.02}$	10.84$_{\pm0.12}$	7.35$_{\pm0.04}$
	STID+MAE	3.50$_{\pm0.02}$	9.34$_{\pm0.10}$	6.70$_{\pm0.04}$	3.60$_{\pm0.02}$	9.52$_{\pm0.07}$	6.82$_{\pm0.04}$	3.70$_{\pm0.02}$	10.51$_{\pm0.08}$	7.12$_{\pm0.04}$	3.78$_{\pm0.02}$	10.86$_{\pm0.08}$	7.37$_{\pm0.04}$
	STID+GATGPT	3.43$_{\pm0.01}$	9.25$_{\pm0.07}$	6.64$_{\pm0.02}$	3.52$_{\pm0.01}$	9.33$_{\pm0.09}$	6.71$_{\pm0.02}$	3.64$_{\pm0.02}$	10.07$_{\pm0.10}$	6.93$_{\pm0.04}$	3.75$_{\pm0.02}$	10.76$_{\pm0.13}$	7.31$_{\pm0.04}$
	STID+Merlin	**3.35**$_{\pm0.01}$	**9.21**$_{\pm0.05}$	**6.58**$_{\pm0.02}$	**3.49**$_{\pm0.01}$	**9.29**$_{\pm0.05}$	**6.65**$_{\pm0.02}$	**3.58**$_{\pm0.02}$	**9.56**$_{\pm0.08}$	**6.81**$_{\pm0.04}$	**3.69**$_{\pm0.02}$	**10.45**$_{\pm0.10}$	**7.06**$_{\pm0.04}$
PEMS04	Corrformer	23.65$_{\pm0.21}$	16.24$_{\pm0.15}$	37.71$_{\pm0.26}$	27.38$_{\pm0.23}$	18.29$_{\pm0.18}$	41.83$_{\pm0.27}$	30.46$_{\pm0.23}$	21.54$_{\pm0.20}$	46.07$_{\pm0.29}$	33.12$_{\pm0.25}$	24.06$_{\pm0.22}$	50.95$_{\pm0.30}$
	MegaCRN	21.95$_{\pm0.18}$	14.82$_{\pm0.13}$	34.06$_{\pm0.22}$	24.43$_{\pm0.20}$	17.15$_{\pm0.14}$	39.48$_{\pm0.24}$	26.09$_{\pm0.22}$	18.49$_{\pm0.17}$	41.18$_{\pm0.25}$	28.29$_{\pm0.24}$	19.91$_{\pm0.20}$	42.81$_{\pm0.26}$
	GPT4TS	22.37$_{\pm0.20}$	14.97$_{\pm0.14}$	35.62$_{\pm0.24}$	25.63$_{\pm0.21}$	18.04$_{\pm0.15}$	39.74$_{\pm0.25}$	27.56$_{\pm0.23}$	19.21$_{\pm0.18}$	42.95$_{\pm0.27}$	29.04$_{\pm0.23}$	20.18$_{\pm0.19}$	44.31$_{\pm0.29}$
	iTransformer+S4	20.64$_{\pm0.16}$	14.08$_{\pm0.14}$	32.56$_{\pm0.19}$	22.76$_{\pm0.18}$	15.34$_{\pm0.16}$	36.25$_{\pm0.21}$	24.34$_{\pm0.19}$	17.26$_{\pm0.15}$	39.16$_{\pm0.23}$	25.94$_{\pm0.21}$	18.06$_{\pm0.18}$	40.23$_{\pm0.24}$
	FourierGNN+SPIN	20.06$_{\pm0.14}$	13.75$_{\pm0.11}$	32.13$_{\pm0.16}$	21.54$_{\pm0.15}$	14.57$_{\pm0.12}$	33.92$_{\pm0.18}$	22.65$_{\pm0.18}$	15.89$_{\pm0.16}$	35.64$_{\pm0.21}$	24.03$_{\pm0.19}$	16.72$_{\pm0.16}$	38.15$_{\pm0.22}$
	DSformer+GATGPT	20.38$_{\pm0.15}$	13.87$_{\pm0.13}$	32.35$_{\pm0.19}$	21.98$_{\pm0.16}$	14.89$_{\pm0.13}$	34.14$_{\pm0.20}$	22.71$_{\pm0.18}$	15.74$_{\pm0.15}$	34.57$_{\pm0.23}$	24.26$_{\pm0.20}$	16.56$_{\pm0.17}$	39.10$_{\pm0.24}$
	TSMixer+GPT2	20.49$_{\pm0.15}$	13.94$_{\pm0.12}$	32.47$_{\pm0.18}$	22.47$_{\pm0.16}$	15.13$_{\pm0.13}$	35.99$_{\pm0.20}$	24.16$_{\pm0.18}$	17.02$_{\pm0.16}$	38.94$_{\pm0.21}$	25.58$_{\pm0.19}$	17.94$_{\pm0.16}$	39.89$_{\pm0.23}$
	STID (Raw)	20.67$_{\pm0.19}$	14.11$_{\pm0.14}$	32.68$_{\pm0.23}$	28.36$_{\pm0.21}$	19.25$_{\pm0.17}$	43.44$_{\pm0.25}$	30.11$_{\pm0.22}$	21.38$_{\pm0.18}$	45.91$_{\pm0.26}$	33.65$_{\pm0.25}$	24.27$_{\pm0.23}$	51.47$_{\pm0.31}$
	STID+SPIN	19.53$_{\pm0.13}$	13.22$_{\pm0.11}$	31.35$_{\pm0.15}$	20.79$_{\pm0.15}$	13.82$_{\pm0.12}$	32.79$_{\pm0.18}$	22.85$_{\pm0.15}$	15.77$_{\pm0.13}$	35.69$_{\pm0.18}$	23.79$_{\pm0.17}$	16.45$_{\pm0.15}$	37.96$_{\pm0.21}$
	STID+GPT2	19.85 $_{\pm0.14}$	13.54$_{\pm0.11}$	31.86$_{\pm0.17}$	21.45$_{\pm0.16}$	14.33$_{\pm0.13}$	33.54$_{\pm0.19}$	22.44$_{\pm0.17}$	15.51$_{\pm0.13}$	35.21$_{\pm0.21}$	23.51$_{\pm0.19}$	16.21$_{\pm0.16}$	37.58$_{\pm0.24}$
	STID+MAE	19.94$_{\pm0.15}$	13.62$_{\pm0.12}$	31.97$_{\pm0.18}$	21.05$_{\pm0.17}$	13.94$_{\pm0.14}$	33.04$_{\pm0.21}$	22.06$_{\pm0.18}$	15.03$_{\pm0.15}$	34.65$_{\pm0.22}$	23.34$_{\pm0.20}$	15.98$_{\pm0.18}$	37.42$_{\pm0.24}$
	STID+GATGPT	19.48$_{\pm0.12}$	13.15$_{\pm0.09}$	31.28$_{\pm0.15}$	20.73$_{\pm0.14}$	14.16$_{\pm0.10}$	32.72$_{\pm0.17}$	21.98$_{\pm0.14}$	14.92$_{\pm0.11}$	35.41$_{\pm0.18}$	23.39$_{\pm0.16}$	16.04$_{\pm0.14}$	37.53$_{\pm0.20}$
	STID+Merlin	**18.86**$_{\pm0.10}$	**12.97**$_{\pm0.07}$	**30.67**$_{\pm0.13}$	**19.56**$_{\pm0.11}$	**13.29**$_{\pm0.09}$	**31.41**$_{\pm0.15}$	**21.19**$_{\pm0.13}$	**14.21**$_{\pm0.11}$	**33.38**$_{\pm0.16}$	**22.62**$_{\pm0.13}$	**15.49**$_{\pm0.12}$	**36.27**$_{\pm0.17}$
China AQI	Corrformer	16.52$_{\pm0.15}$	34.96$_{\pm0.21}$	27.81$_{\pm0.20}$	18.32$_{\pm0.16}$	39.27$_{\pm0.22}$	30.44$_{\pm0.21}$	20.47$_{\pm0.19}$	43.51$_{\pm0.24}$	31.95$_{\pm0.22}$	22.48$_{\pm0.23}$	45.37$_{\pm0.28}$	34.79$_{\pm0.26}$
	MegaCRN	16.35$_{\pm0.15}$	34.75$_{\pm0.21}$	27.61$_{\pm0.20}$	18.14$_{\pm0.16}$	38.43$_{\pm0.22}$	29.46$_{\pm0.20}$	19.96$_{\pm0.18}$	42.64$_{\pm0.23}$	32.54$_{\pm0.21}$	22.06$_{\pm0.21}$	44.28$_{\pm0.27}$	34.42$_{\pm0.24}$
	GPT4TS	16.03$_{\pm0.15}$	33.06$_{\pm0.21}$	27.04$_{\pm0.20}$	17.85$_{\pm0.16}$	37.68$_{\pm0.22}$	28.91$_{\pm0.21}$	19.28$_{\pm0.18}$	41.15$_{\pm0.24}$	32.07$_{\pm0.21}$	21.65$_{\pm0.21}$	43.97$_{\pm0.26}$	33.95$_{\pm0.25}$
	iTransformer+S4	15.49$_{\pm0.13}$	32.06$_{\pm0.19}$	25.57$_{\pm0.17}$	16.79$_{\pm0.15}$	35.76$_{\pm0.21}$	27.84$_{\pm0.19}$	18.44$_{\pm0.17}$	39.76$_{\pm0.22}$	30.68$_{\pm0.21}$	21.32$_{\pm0.20}$	43.62$_{\pm0.25}$	33.68$_{\pm0.23}$
	FourierGNN+SPIN	15.28$_{\pm0.12}$	31.44$_{\pm0.18}$	25.24$_{\pm0.15}$	16.17$_{\pm0.14}$	34.13$_{\pm0.20}$	27.02$_{\pm0.17}$	18.05$_{\pm0.15}$	38.56$_{\pm0.21}$	30.06$_{\pm0.19}$	20.53$_{\pm0.17}$	42.15$_{\pm0.22}$	32.43$_{\pm0.20}$
	DSformer+GATGPT	15.39$_{\pm0.12}$	31.89$_{\pm0.18}$	25.43$_{\pm0.16}$	16.39$_{\pm0.14}$	34.82$_{\pm0.20}$	27.58$_{\pm0.19}$	18.29$_{\pm0.16}$	39.37$_{\pm0.22}$	30.17$_{\pm0.20}$	21.07$_{\pm0.18}$	42.97$_{\pm0.22}$	33.04$_{\pm0.21}$
	TSMixer+GPT2	15.45$_{\pm0.12}$	32.04$_{\pm0.18}$	25.59$_{\pm0.16}$	16.43$_{\pm0.14}$	34.73$_{\pm0.20}$	27.65$_{\pm0.18}$	18.33$_{\pm0.16}$	39.85$_{\pm0.22}$	30.23$_{\pm0.20}$	21.25$_{\pm0.18}$	43.54$_{\pm0.23}$	33.59$_{\pm0.21}$
	STID (Raw)	15.53$_{\pm0.14}$	32.46$_{\pm0.20}$	25.71$_{\pm0.19}$	18.56$_{\pm0.16}$	39.95$_{\pm0.22}$	30.47$_{\pm0.21}$	20.36$_{\pm0.19}$	43.63$_{\pm0.25}$	32.09$_{\pm0.22}$	23.24$_{\pm0.21}$	46.18$_{\pm0.26}$	35.54$_{\pm0.24}$
	STID+SPIN	14.98$_{\pm0.09}$	30.25$_{\pm0.16}$	25.06$_{\pm0.13}$	15.67$_{\pm0.13}$	32.25$_{\pm0.19}$	25.98$_{\pm0.17}$	17.43$_{\pm0.15}$	37.65$_{\pm0.21}$	28.89$_{\pm0.19}$	19.94$_{\pm0.17}$	41.78$_{\pm0.23}$	32.16$_{\pm0.20}$
	STID+GPT2	15.12$_{\pm0.12}$	30.89$_{\pm0.17}$	25.15$_{\pm0.15}$	15.89$_{\pm0.14}$	32.84$_{\pm0.20}$	26.74$_{\pm0.18}$	17.35$_{\pm0.15}$	37.22$_{\pm0.20}$	28.72$_{\pm0.18}$	19.50$_{\pm0.16}$	41.26$_{\pm0.22}$	31.73$_{\pm0.19}$
	STID+MAE	15.22$_{\pm0.12}$	31.06$_{\pm0.18}$	25.19$_{\pm0.16}$	15.94$_{\pm0.14}$	32.76$_{\pm0.20}$	26.97$_{\pm0.18}$	17.29$_{\pm0.14}$	37.05$_{\pm0.19}$	28.42$_{\pm0.17}$	19.23$_{\pm0.15}$	40.53$_{\pm0.21}$	31.59$_{\pm0.18}$
	STID+GATGPT	15.07$_{\pm0.10}$	30.53$_{\pm0.16}$	25.11$_{\pm0.14}$	15.75$_{\pm0.12}$	32.65$_{\pm0.18}$	26.61$_{\pm0.16}$	17.26$_{\pm0.13}$	36.94$_{\pm0.19}$	29.15$_{\pm0.17}$	19.19$_{\pm0.15}$	40.56$_{\pm0.20}$	31.36$_{\pm0.18}$
	STID+Merlin	**14.89**$_{\pm0.08}$	**29.97**$_{\pm0.15}$	**24.93**$_{\pm0.12}$	**15.39**$_{\pm0.10}$	**31.86**$_{\pm0.16}$	**25.46**$_{\pm0.14}$	**16.83**$_{\pm0.11}$	**36.30**$_{\pm0.17}$	**27.30**$_{\pm0.15}$	**18.68**$_{\pm0.13}$	**39.39**$_{\pm0.19}$	**30.31**$_{\pm0.17}$
Global Wind	Corrformer	5.78$_{\pm0.02}$	34.32$_{\pm0.17}$	8.52$_{\pm0.04}$	5.99$_{\pm0.02}$	37.18$_{\pm0.19}$	8.79$_{\pm0.05}$	6.29$_{\pm0.04}$	42.65$_{\pm0.20}$	9.18$_{\pm0.07}$	6.59$_{\pm0.04}$	45.98$_{\pm0.22}$	9.63$_{\pm0.08}$
	MegaCRN	5.71$_{\pm0.02}$	32.98$_{\pm0.16}$	8.39$_{\pm0.03}$	5.91$_{\pm0.02}$	36.12$_{\pm0.18}$	8.71$_{\pm0.04}$	6.17$_{\pm0.04}$	40.69$_{\pm0.19}$	9.09$_{\pm0.07}$	6.44$_{\pm0.04}$	45.21$_{\pm0.21}$	9.48$_{\pm0.08}$
	GPT4TS	5.73$_{\pm0.02}$	33.25$_{\pm0.16}$	8.41$_{\pm0.03}$	5.95$_{\pm0.02}$	36.57$_{\pm0.18}$	8.76$_{\pm0.04}$	6.23$_{\pm0.04}$	41.35$_{\pm0.20}$	9.13$_{\pm0.07}$	6.53$_{\pm0.04}$	45.79$_{\pm0.21}$	9.56$_{\pm0.08}$
	iTransformer+S4	5.62$_{\pm0.01}$	32.66$_{\pm0.15}$	8.30$_{\pm0.02}$	5.86$_{\pm0.02}$	35.12$_{\pm0.17}$	8.67$_{\pm0.04}$	6.10$_{\pm0.02}$	39.45$_{\pm0.18}$	8.94$_{\pm0.05}$	6.32$_{\pm0.04}$	43.61$_{\pm0.20}$	9.24$_{\pm0.07}$
	FourierGNN+SPIN	5.59$_{\pm0.01}$	32.18$_{\pm0.14}$	8.23$_{\pm0.02}$	5.72$_{\pm0.02}$	33.22$_{\pm0.16}$	8.43$_{\pm0.03}$	5.95$_{\pm0.02}$	35.69$_{\pm0.17}$	8.69$_{\pm0.04}$	6.16$_{\pm0.03}$	40.18$_{\pm0.18}$	9.01$_{\pm0.06}$
	DSformer+GATGPT	5.60$_{\pm0.01}$	32.25$_{\pm0.13}$	8.25$_{\pm0.02}$	5.79$_{\pm0.02}$	34.53$_{\pm0.16}$	8.54$_{\pm0.03}$	5.98$_{\pm0.02}$	37.21$_{\pm0.17}$	8.76$_{\pm0.04}$	6.21$_{\pm0.03}$	41.25$_{\pm0.18}$	9.15$_{\pm0.06}$
	TSMixer+GPT2	5.61$_{\pm0.01}$	32.58$_{\pm0.14}$	8.28$_{\pm0.02}$	5.83$_{\pm0.02}$	34.94$_{\pm0.16}$	8.62$_{\pm0.03}$	6.09$_{\pm0.02}$	38.52$_{\pm0.17}$	8.91$_{\pm0.05}$	6.31$_{\pm0.03}$	43.57$_{\pm0.18}$	9.22$_{\pm0.06}$
	STID (Raw)	5.63$_{\pm0.01}$	32.73$_{\pm0.15}$	8.31$_{\pm0.02}$	6.05$_{\pm0.02}$	38.49$_{\pm0.18}$	8.87$_{\pm0.04}$	6.34$_{\pm0.04}$	43.19$_{\pm0.19}$	9.25$_{\pm0.06}$	6.68$_{\pm0.04}$	46.72$_{\pm0.22}$	9.77$_{\pm0.08}$
	STID+SPIN	5.53$_{\pm0.01}$	31.15$_{\pm0.11}$	7.93$_{\pm0.02}$	5.64$_{\pm0.01}$	32.78$_{\pm0.14}$	8.33$_{\pm0.02}$	5.97$_{\pm0.02}$	36.71$_{\pm0.17}$	8.74$_{\pm0.04}$	6.22$_{\pm0.03}$	41.45$_{\pm0.18}$	9.11$_{\pm0.07}$
	STID+GPT2	5.57$_{\pm0.01}$	32.01$_{\pm0.12}$	7.99$_{\pm0.02}$	5.69$_{\pm0.02}$	33.09$_{\pm0.15}$	8.39$_{\pm0.03}$	5.89$_{\pm0.02}$	35.90$_{\pm0.17}$	8.65$_{\pm0.04}$	6.15$_{\pm0.03}$	40.05$_{\pm0.18}$	9.08$_{\pm0.06}$
	STID+MAE	5.58$_{\pm0.01}$	32.06$_{\pm0.13}$	8.04$_{\pm0.02}$	5.71$_{\pm0.02}$	33.25$_{\pm0.15}$	8.43$_{\pm0.04}$	5.86$_{\pm0.02}$	35.46$_{\pm0.16}$	8.62$_{\pm0.04}$	6.11$_{\pm0.02}$	39.45$_{\pm0.17}$	9.02$_{\pm0.05}$
	STID+GATGPT	5.55$_{\pm0.01}$	31.75$_{\pm0.11}$	7.98$_{\pm0.02}$	5.68$_{\pm0.01}$	32.45$_{\pm0.13}$	8.37$_{\pm0.02}$	5.85$_{\pm0.02}$	35.08$_{\pm0.16}$	8.57$_{\pm0.04}$	6.13$_{\pm0.02}$	39.84$_{\pm0.17}$	9.05$_{\pm0.05}$
	STID+Merlin	**5.49**$_{\pm0.01}$	**30.54**$_{\pm0.10}$	**7.85**$_{\pm0.02}$	**5.57**$_{\pm0.01}$	**31.98**$_{\pm0.12}$	**8.01**$_{\pm0.02}$	**5.78**$_{\pm0.01}$	**34.19**$_{\pm0.14}$	**8.49**$_{\pm0.02}$	**6.02**$_{\pm0.02}$	**38.47**$_{\pm0.16}$	**8.84**$_{\pm0.04}$

4.3 TRANSFERABILITY OF MERLIN

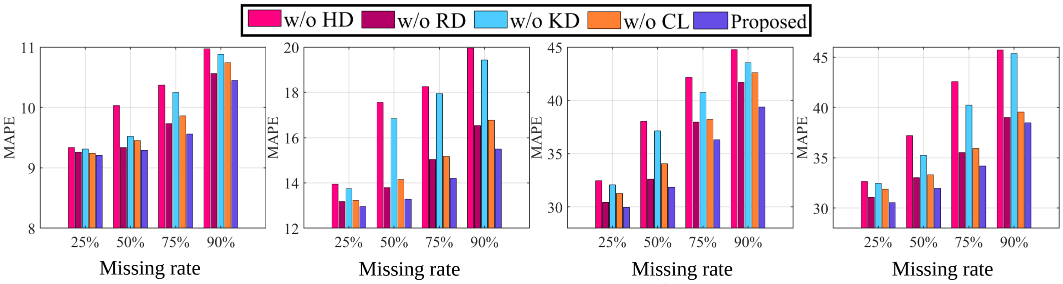
It can be found from the main results that Merlin can effectively improve the forecasting performance of STID in MTSF with sparse observations. To further validate the effectiveness and transferability of Merlin, we choose three other models (TSmixer, DSformer, and FourierGNN) as backbones and compare the performance of Merlin with other imputation methods (GATGPT, GPT2, MAE and SPIN). Table 3 shows the MAE values of Merlin and other imputation methods. Based on the results, we can draw the following conclusions: (1) Advanced one-stage models struggle to perform well in MTSF with sparse observations. Specifically, the presence of missing data makes it difficult for existing models to mine semantics from sparse observations, resulting in poor robustness. Therefore, existing forecasting models struggle to achieve satisfactory results. (2) Compared with SPIN, the generative imputation methods can achieve better forecasting results when the missing rate is higher. The main reason is that SPIN relies on local spatial-temporal information, which makes its performance limited at high missing rates. (3) Compared with other methods, Merlin can better restore the performance of all backbone models on all datasets. The experimental results fully prove the transfer ability and practical value of Merlin. Specifically, Merlin can help existing advanced models achieve semantic alignment between sparse observations and complete observations, thereby effectively enhancing the model's robustness and achieving better forecasting results.

4.4 ABLATION EXPERIMENTS

We conduct ablation experiments from the following perspectives: (1) **w/o HD**: We remove the hidden representation distillation. (2) **w/o RD**: We remove the forecasting result distillation. (3) **w/o KD**: We removed the teacher model and knowledge distillation. In this case, STID uses complete observations and sparse observations to construct contrastive learning. (4) **w/o CL**: We remove the multi-view contrastive learning. Figure 3 shows the results of the ablation experiment. Based on the experimental results, we can draw the following conclusions: (1) The forecasting result distillation

Table 3: MAE values of Merlin and other methods (The best results are shown in **bold**).

Backbone	Methods	METR-LA				PEMS04				China AQI				Global Wind			
		25%	50%	75%	90%	25%	50%	75%	90%	25%	50%	75%	90%	25%	50%	75%	90%
TSmixer	+Merlin	**3.44**	**3.54**	**3.66**	**3.78**	**19.53**	**21.54**	**22.39**	**23.95**	**15.18**	**16.07**	**17.94**	**20.58**	**5.55**	**5.77**	**5.96**	**6.15**
	+GATGPT	3.46	3.59	3.69	3.81	19.97	21.85	23.06	24.47	15.22	16.38	18.25	20.97	5.58	5.79	6.03	6.22
	+GPT2	3.48	3.62	3.71	3.85	20.49	22.47	24.16	25.58	15.45	16.43	18.33	21.25	5.61	5.83	6.09	6.31
	+MAE	3.53	3.65	3.73	3.84	20.67	22.58	24.23	25.37	15.52	16.52	18.27	21.03	5.64	5.92	6.11	6.28
	+SPIN	3.47	3.60	3.72	3.87	20.23	22.15	24.19	25.76	15.25	16.39	18.41	21.54	5.60	5.81	6.07	6.34
	raw	3.62	3.78	3.95	4.06	21.53	26.39	29.18	31.42	16.33	18.44	20.59	22.98	5.77	6.01	6.31	6.63
DSformer	+Merlin	**3.49**	**3.61**	**3.70**	**3.82**	**20.17**	**21.67**	**22.08**	**23.84**	**15.23**	**16.15**	**18.07**	**20.78**	**5.54**	**5.72**	**5.87**	**6.14**
	+GATGPT	3.52	3.65	3.78	3.89	20.38	21.98	22.71	24.26	15.39	16.72	18.76	21.35	5.60	5.79	5.98	6.21
	+GPT2	3.56	3.69	3.83	3.97	20.79	22.59	23.78	25.14	15.54	16.82	18.84	21.46	5.62	5.82	6.04	6.25
	+MAE	3.57	3.71	3.85	3.95	20.94	22.67	23.84	24.98	15.87	16.91	18.90	21.39	5.68	5.89	6.05	6.23
	+SPIN	3.54	3.66	3.82	3.98	20.54	22.45	23.95	25.47	15.43	16.74	18.79	21.54	5.64	5.78	6.01	6.27
	raw	3.72	3.87	3.95	4.11	23.24	27.85	30.47	33.25	16.52	18.75	20.96	23.47	5.75	5.98	6.25	6.57
FourierGNN	+Merlin	**3.45**	**3.53**	**3.65**	**3.76**	**19.32**	**20.19**	**21.76**	**23.24**	**15.04**	**15.92**	**17.67**	**20.04**	**5.52**	**5.67**	**5.88**	**6.06**
	+GATGPT	3.48	3.57	3.68	3.79	19.76	20.86	22.13	23.51	15.19	16.15	17.97	20.37	5.56	5.69	5.91	6.10
	+GPT2	3.53	3.61	3.72	3.84	19.97	21.61	22.58	23.91	15.37	16.25	18.12	20.51	5.58	5.71	5.94	6.14
	+MAE	3.55	3.66	3.75	3.83	20.08	21.73	22.70	23.87	15.42	16.31	18.15	20.49	5.61	5.73	5.93	6.12
	+SPIN	3.50	3.58	3.71	3.86	19.83	21.54	22.65	24.03	15.28	16.17	18.05	20.53	5.59	5.72	5.95	6.16
	raw	3.61	3.77	3.92	4.05	21.34	24.58	27.05	29.71	15.98	17.69	19.13	21.57	5.73	5.93	6.15	6.39



(a) METR-LA **(b) PEMS04** **(c) China AQI** **(d) Global Wind**

Figure 3: Results of ablation experiments.

has the least effect on the results. The experimental results show that as long as the encoder can mine important semantics, the decoder can realize effective forecasting. (2) When the missing rate is large, the effect of multi-view contrastive learning increases significantly. The main reason is that the STID has the ability to mine semantics when the missing rate is low. (3) When STID does not use the teacher model and knowledge distillation, it can only use contrastive learning to help STID learn how to align the semantics between sparse observations and complete observations. In this case, without the guidance of teachers, it is difficult for STID to fully mine semantics from sparse observations. (4) After the hidden representation distillation is removed, the forecasting performance of STID decreases significantly. The main reason is that hidden representation distillation enables STID to learn how to make full use of sparse observations to obtain representations that can be obtained with complete observations, which is crucial for aligning the semantics between sparse observations and complete observations.

5 CONCLUSION

This paper considers the challenge of MTSF with unfixed missing rates from the perspective of robustness. Specifically, existing models face two challenges when modeling sparse observations: on the one hand, they must address the issue of missing values disrupting the semantics of MTS. On the other hand, they also need to face the challenge that the missing rate of MTS is unfixed at different time points in the real world. To this end, we propose Merlin based on offline knowledge distillation and multi-view contrastive learning. Merlin aims to assist existing models in effectively achieving semantic alignment between sparse observations with different missing rates and complete observations, thereby significantly enhancing their robustness. Extensive experiments show that the proposed model achieves satisfactory forecasting results on all datasets and settings. Additionally, Merlin can significantly improve the performance and robustness of existing forecasting models in MTSF with unfixed missing rates. In future work, we plan to investigate the effects of knowledge distillation when the teacher model and the student model utilize different network structures, such as large language models.

REFERENCES

Hyun Ahn, Kyunghee Sun, and Kwanghoon Pio Kim. Comparison of missing data imputation methods in time series forecasting. *Computers, Materials & Continua*, 70(1):767–779, 2022.

Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, Yuyang Wang, Danielle Maddix, Caner Turkmen, Jan Gasthaus, Michael Bohlke-Schneider, David Salinas, Lorenzo Stella, et al. Deep learning for time series forecasting: Tutorial and literature survey. *ACM Computing Surveys*, 55(6):1–36, 2022.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023.

Muhammad Ali Chattha, Ludger van Elst, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. Kenn: Enhancing deep neural networks by leveraging knowledge for time series forecasting. *arXiv preprint arXiv:2202.03903*, 2022.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Xiaodan Chen, Xiucheng Li, Bo Liu, and Zhijun Li. Biased temporal convolution graph network for time series forecasting with missing values. In *The Twelfth International Conference on Learning Representations*, 2023b.

Yakun Chen, Xianzhi Wang, and Guandong Xu. Gatgpt: A pre-trained large language model with graph attention network for spatiotemporal imputation. *arXiv preprint arXiv:2311.14332*, 2023c.

Yu Chengqing, Yan Guangxi, Yu Chengming, Zhang Yu, and Mi Xiwei. A multi-factor driven spatiotemporal wind power prediction model based on ensemble deep graph attention reinforcement learning networks. *Energy*, 263:126034, 2023.

Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. In *International Conference on Learning Representations*, 2022.

Aimei Dong, Jian Liu, Guodong Zhang, Zhonghe Wei, Yi Zhai, and Guohua Lv. Momentum contrast transformer for covid-19 diagnosis with knowledge distillation. *Pattern Recognition*, 143:109732, 2023.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

Wenjie Du, David Côté, and Yan Liu. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023.

Jianping Gou, Liyuan Sun, Baosheng Yu, Shaohua Wan, and Dacheng Tao. Hierarchical multi-attention transfer for knowledge distillation. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(2):1–20, 2023.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *The Tenth International Conference on Learning Representations*, 2022.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pp. 4116–4126. PMLR, 2020.

Marisca Ivan, Cini Andrea, and Cesare Alippi. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, pp. 1–17, 2022.

Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8078–8086, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 4189–4196, 2021.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

Zhe Li, Zhongwen Rao, Lujia Pan, Pengyun Wang, and Zenglin Xu. Ti-mae: Self-supervised masked time series autoencoders. *arXiv preprint arXiv:2301.08871*, 2023.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021.

Hui Liu, Chengqing Yu, Haiping Wu, Zhu Duan, and Guangxi Yan. A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting. *Energy*, 202:117794, 2020.

Jiexi Liu and Songcan Chen. Timesurl: Self-supervised contrastive learning for universal time series representation learning. *arXiv preprint arXiv:2312.15709*, 2023.

Xinwei Liu, Muchuan Qin, Yue He, Xiwei Mi, and Chengqing Yu. A new multi-data-driven spatiotemporal pm2. 5 forecasting model based on an ensemble graph reinforcement learning convolutional network. *Atmospheric Pollution Research*, 12(10):101197, 2021.

Xu Liu, Yuxuan Liang, Chao Huang, Yu Zheng, Bryan Hooi, and Roger Zimmermann. When do contrastive learning signals help spatio-temporal graph forecasting? In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pp. 1–12, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Ivan Marisca, Cesare Alippi, and Filippo Maria Bianchi. Graph-based forecasting with missing data through spatiotemporal downsampling. In *Forty-first International Conference on Machine Learning*, 2024.

Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8983–8991, 2021.

Alessio Monti, Angelo Porrello, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. How many observations are enough? knowledge distillation for trajectory forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6553–6562, 2022.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. Xtremedistil: Multi-stage distillation for massive multilingual models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2221–2234, 2020.

Siyuan Shan, Yang Li, and Junier B Oliva. Nrtsi: Non-recurrent time series imputation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pp. 4454–4458, 2022a.

Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment*, 15(11):2733–2746, 2022b.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Guangyin Jin, Xin Cao, Gao Cong, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *arXiv preprint arXiv:2310.06119*, 2023.

S Sridevi, S Rajaram, C Parthiban, S SibiArasan, and C Swadhikar. Imputation for the analysis of missing values and prediction of time series data. In *2011 international conference on recent trends in information Technology (ICRTIT)*, pp. 1158–1163. IEEE, 2011.

Jing Tan, Hui Liu, Yanfei Li, Shi Yin, and Chengqing Yu. A new ensemble spatio-temporal pm2. 5 prediction method based on graph attention recursive networks and reinforcement learning. *Chaos, Solitons & Fractals*, 162:112405, 2022.

Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5956–5963, 2020.

Trang H Tran, Lam M Nguyen, Kyongmin Yeo, Nam Nguyen, Dzung Phan, Roman Vaculin, and Jayant Kalagnanam. An end-to-end time series model for simultaneous imputation and forecast. *arXiv preprint arXiv:2306.00778*, 2023.

Fei Wang, Di Yao, Yong Li, Tao Sun, and Zhao Zhang. Ai-enhanced spatial-temporal data-mining technology: New chance for next-generation urban computing. *The Innovation*, 4(2), 2023.

Kai Wang, Yu Liu, Qian Ma, and Quan Z Sheng. Mulde: Multi-teacher knowledge distillation for low-dimensional knowledge graph embeddings. In *Proceedings of the Web Conference 2021*, pp. 1716–1726, 2021.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023a.

Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, pp. 1–10, 2023b.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019.

Qing Xu, Zhenghua Chen, Mohamed Ragab, Chao Wang, Min Wu, and Xiaoli Li. Contrastive adversarial knowledge distillation for deep model compression in time-series regression tasks. *Neurocomputing*, 485:242–251, 2022.

Yi Xu, Armin Bazarjani, Hyung-gun Chi, Chiho Choi, and Yun Fu. Uncovering the missing pattern: Unified framework towards trajectory imputation and prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9632–9643, 2023.

Yongjun Xu, Xin Liu, Xin Cao, Changping Huang, Enke Liu, Sen Qian, Xingchen Liu, Yanjun Wu, Fengliang Dong, Cheng-Wei Qiu, et al. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, 2(4):100179, 2021.

Chuanguang Yang, Helong Zhou, Zhulin An, Xue Jiang, Yongjun Xu, and Qian Zhang. Cross-image relational knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12319–12328, 2022.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Chengqing Yu, Fei Wang, Zezhi Shao, Tao Sun, Lin Wu, and Yongjun Xu. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3062–3072, 2023a.

Chengqing Yu, Guangxi Yan, Chengming Yu, and Xiwei Mi. Attention mechanism is useful in spatio-temporal wind speed prediction: Evidence from china. *Applied Soft Computing*, 148: 110864, 2023b.

Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, and Yongjun Xu. Ginar: An end-to-end multivariate time series forecasting model suitable for variable missing. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3989–4000, 2024a.

Chengqing Yu, Fei Wang, Yilun Wang, Zezhi Shao, Tao Sun, Di Yao, and Yongjun Xu. Mgsfformer: A multi-granularity spatiotemporal fusion transformer for air quality prediction. *Information Fusion*, pp. 102607, 2024b. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2024.102607.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8980–8987, 2022.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2114–2124, 2021.

Xiaoxia Zhang, Shang Shi, HaiChao Sun, Degang Chen, Guoyin Wang, and Kesheng Wu. Acvae: A novel self-adversarial variational auto-encoder combined with contrast learning for time series anomaly detection. *Neural Networks*, 171:383–395, 2024.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2022.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1234–1241, 2020.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, Jianzhong Qi, Chaochao Chen, and Longbiao Chen. Increase: Inductive graph representation learning for spatio-temporal kriging. In *Proceedings of the ACM Web Conference 2023*, pp. 673–683, 2023.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

A IMPLEMENTATION DETAILS

A.1 DATASETS

The basic statistics for these datasets are shown in Table 4. A brief introduction to these datasets is provided as follows:

- **METR-LA**[2]: It is a traffic speed dataset collected by loop-detectors located on the LA County road network, which contains data collected by 207 sensors from Mar 1st, 2012 to Jun 30th, 2012. Each time series is sampled at a 5-minute interval, totaling 34272 time slices.
- **PEMS04**[3]: It is a traffic flow dataset collected by CalTrans PeMS, which contains data collected by 307 sensors from January 1st, 2018, to February 28th, 2018. Each time series is sampled at a 5-minute interval, totaling 16992 time slices.
- **China AQI**[4]: It is an air quality dataset collected by environmental monitoring stations in China, which includes data from 1,300 air monitoring stations from January 2015 to December 2020. Each time series is sampled at a 1-hour interval, totaling 41,506 time slices.
- **Global Wind**[5]: It is derived from the global wind speed dataset of the National Oceanic and Atmospheric Administration (NOAA) National Center for Environmental Information (NCEI), which includes data from 2,908 meteorological monitoring stations from 1993 to 2022. Each time series is sampled at a 1-day interval, totaling 10,957 time slices.

Table 4: The statistics of four datasets.

Datasets	Variates	Timesteps	Granularity
METR-LA	207	34272	5 minutes
PEMS04	307	16992	5 minutes
China AQI	1300	41506	1 hour
Global Wind	2908	10957	1 day

A.2 BASELINES

The hyperparameter settings for the baselines are selected based on their original papers and codes. The search process of hyperparameters is mainly based on the grid search method. All baselines are introduced as follows:

- **Corrformer**: It uses autoregressive attention and cross attention to mine spatial-temporal correlations.
- **MegaCRN**: It uses utilizes the memory bank to enhance the adaptive graph convolution's ability to model spatial correlations and embeds the component into the recurrent neural network.
- **GPT4TS**: It uses a pretrained GPT2 to encode the context of time series, and then employs a linear decoder to obtain the forecasting results.
- **STID**: It uses spatial-temporal identity embedding to improve the ability of MLP to mine multivariate time series.
- **STID+SPIN**: SPIN effectively combines temporal attention, spatial attention, and cross attention to mine the spatial-temporal correlation of multivariate time series, thereby improving the effectiveness of data recovery.
- **STID+GPT2**: It first uses GPT2 to recover missing values, and then uses STID to model the processed data.

---

[2] https://github.com/liyaguang/DCRNN

[3] https://github.com/guoshnBJTU/ASTGNN/tree/main/data

[4] https://quotsoft.net/air/

[5] https://www.ncei.noaa.gov/

- **STID+MAE**: MAE adopts autoencoder structure to improve the effect of data recovery.
- **STID+GATGPT**: GATGPT combines GPT and graph attention mechanism to recover missing data by fully using spatial-temporal correlations.
- **iTransformer+S4**: iTransformer changes the function of the attention and feedforward layer to improve the time series forecasting results. S4 uses the fundamental state space model to mine temporal information of time series.
- **FourierGNN+SPIN**: FourierGNN uses Fourier Graph Operator to replace GCN and obtain better time series forecasting results.
- **DSformer+GATGPT**: DSformer uses uses double sampling block and temporal variable attention block to realize multivariate time series forecasting.
- **TSMixer+GPT2**: TSMixer uses residual connections and MLP to mine spatial-temporal correlations. Compared with complex models, this framework has the advantages of both performance and efficiency.

B HYPERPARAMETER ANALYSIS

Table 5 shows the main hyperparameters of the backbone (STID) and Merlin. We evaluate three hyperparameters that have the greatest impact on Merlin (The weight of the loss, batch size and temperature parameter) (Chen et al. (2020)). Besides, we also evaluate three hyperparameters that have the greatest impact on the backbone (Embeding size, input length and number of layers).

The experimental results of hyperparameter analysis are shown in Figure 4 to Figure 7. Based on the hyperparameter analysis results, we can draw the following conclusions: (1) Appropriately increasing the batch size can improve the forecasting accuracy of STID. On the one hand, the increase of batch size can increase the number of negative data pairs, which can better enhance the model's robustness and uncover key semantic information. On the other hand, too large batch size can lead to premature convergence of STID, resulting in underfitting problems. (2) Proper balance of temperature parameter is important to improve the effect of contrastive learning. On the one hand, properly reducing the temperature parameter can improve the effect of the model and improve convergence. On the other hand, the value of temperature parameter being too small may lead to the problem of local optimality. (3) When the weight of the loss is set to 1, the proposed model can perform best, which fully demonstrates the importance of Merlin. Specifically, the proposed loss functions help STID realize semantic alignment effectively, reduce the interference of missing values, and thus guarantee the forecasting performance. (4) Properly balancing the size of the embedding dimension and the number of layers can effectively ensure the forecasting performance of STID. Specifically, too few parameters fail to sufficiently exploit the sparse observations, while too many parameters can lead to overfitting. (5) The input length has a significant impact on the forecasting results. The main reason is that the input length determines the amount of information that the model can capture. If the input length is too short, it fails to provide sufficient useful information, whereas an excessively long input length can lead to overfitting.

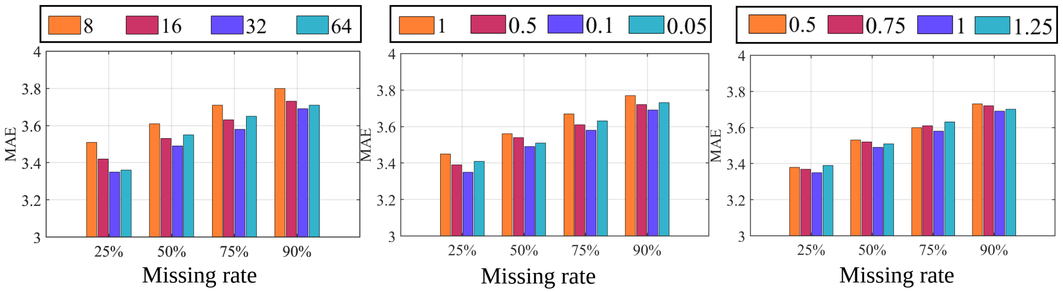
C EFFICIENCY

In order to demonstrate the efficiency advantages of Merlin, this section compares the training times on the PEMS04 dataset for STID+Merlin, STID+GPT2, STID+GATGPT, iTransformer+S4, and FourierGNN+SPIN. Specifically, considering that STID+Merlin only needs to be trained once to adapt to different missing rates, whereas the other baselines require separate training sessions for each missing rate, we directly recorded the training time of STID+Merlin for a single epoch and summed up the training times for each missing rate for the other baselines. The experimental equipment is the Intel(R) Xeon(R) Gold 5217 CPU @ 3.00GHz, 128G RAM computing server with RTX 3090 graphics card.

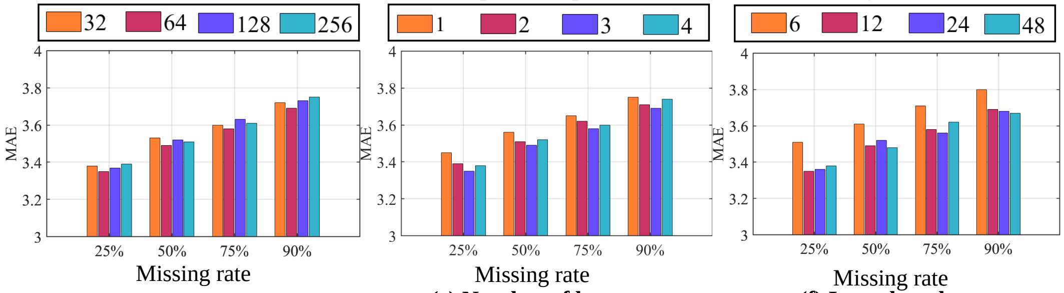
Figure 8 displays the average training time per epoch for these models. Based on the experimental results, the following conclusions can be drawn: (1) Compared to two-stage models, STID+Merlin requires less training time. The main reason is that STID+Merlin only needs to train one teacher model and one student model. (2) Since neither the imputation model nor the teacher model is

Table 5: Values of the corresponding hyperparameters.

Methods	Config	Values
Merlin	batch size	32
	β	1
	temperature parameter	0.1
STID	optimizer	Adam (Kingma & Ba (2014))
	learning rate	0.002
	embeding size	64
	node embeding size	64
	temporal embeding size (day)	64
	temporal embeding size (week)	64
	number of layers	3
	dropout	0.15
	learning rate schedule	MultiStepLR
	clip gradient normalization	5
	milestone	[1, 50, 80]
	gamme	0.5
	epoch	100

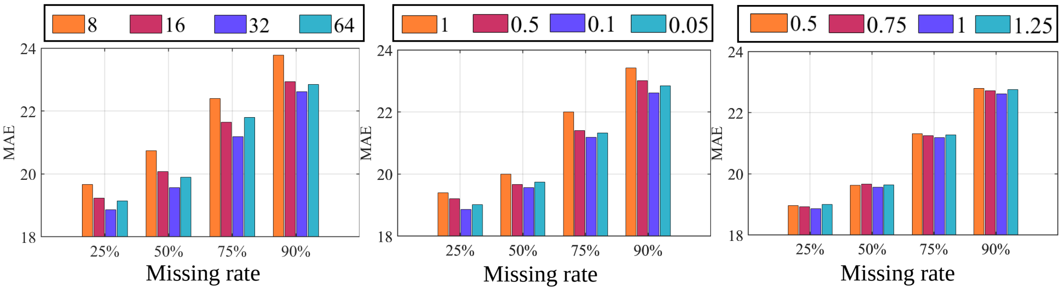


(a) Batch size **(b) Temperature parameter** **(c) The weight of the Loss**

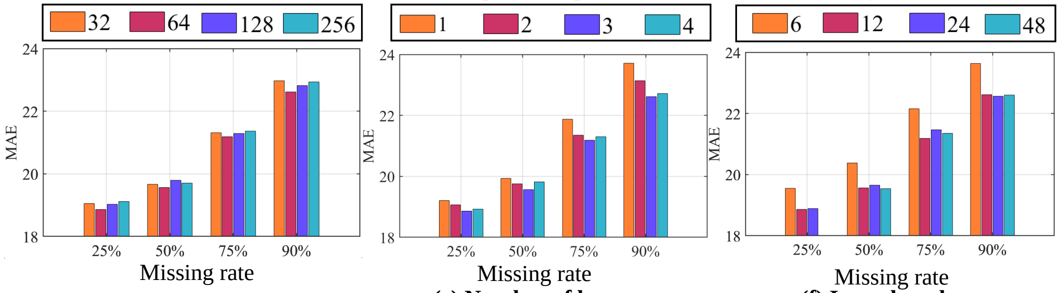


(d) Embedding size **(e) Number of layers** **(f) Input length**

Figure 4: The results of hyperparameter experiment (METR-LA dataset).

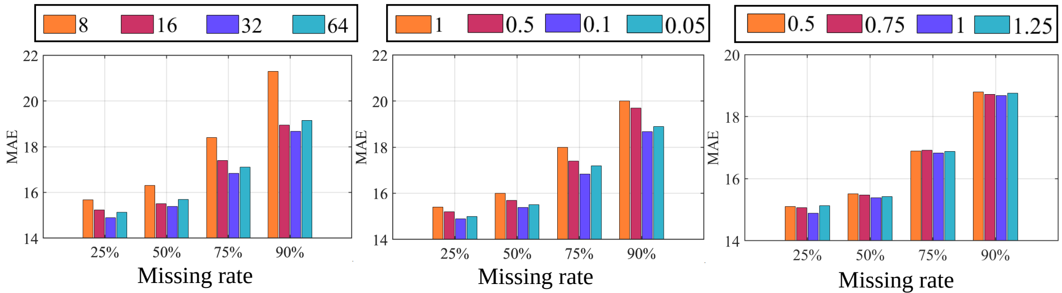


(a) Batch size **(b) Temperature parameter** **(c) The weight of the Loss**

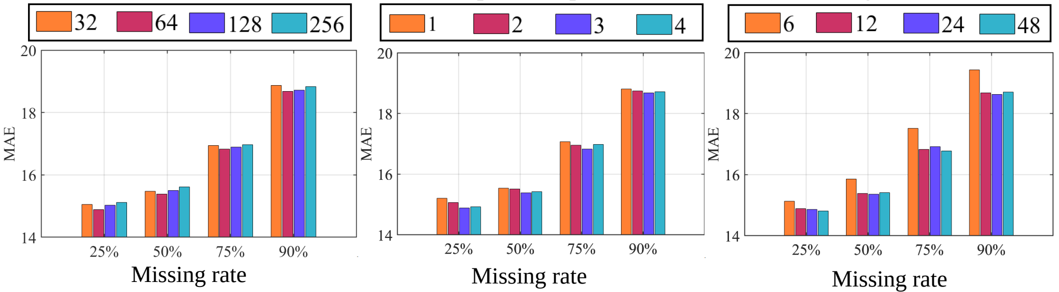


(d) Embedding size **(e) Number of layers** **(f) Input length**

Figure 5: The results of hyperparameter experiment (PEMS04 dataset).

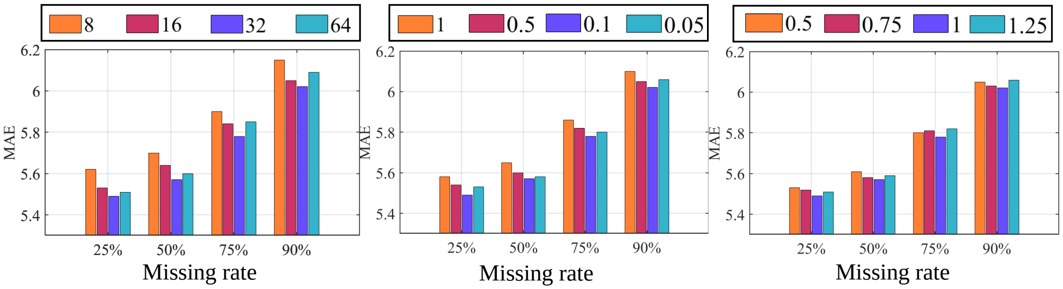


(a) Batch size **(b) Temperature parameter** **(c) The weight of the Loss**

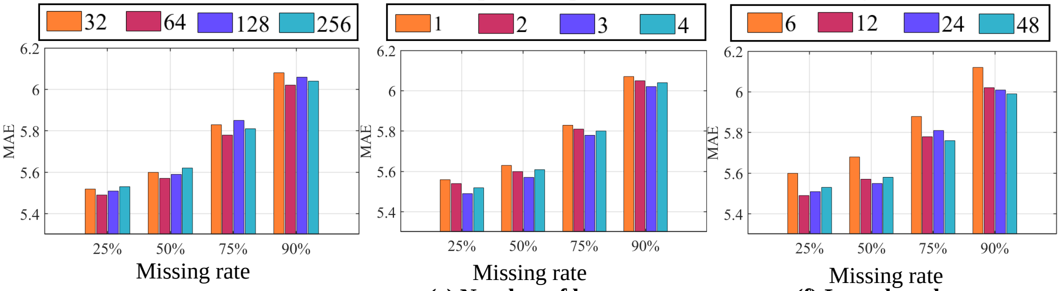


(d) Embedding size **(e) Number of layers** **(f) Input length**

Figure 6: The results of hyperparameter experiment (China AQI dataset).



(a) Batch size **(b) Temperature parameter** **(c) The weight of the Loss**



(d) Embedding size **(e) Number of layers** **(f) Input length**

Figure 7: The results of hyperparameter experiment (Global Wind dataset).

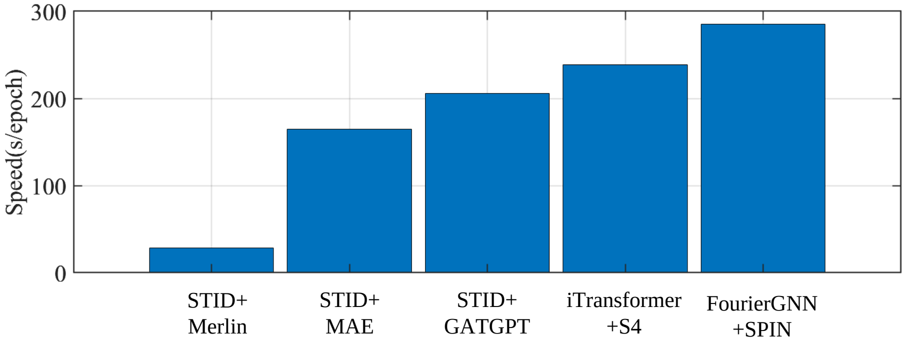
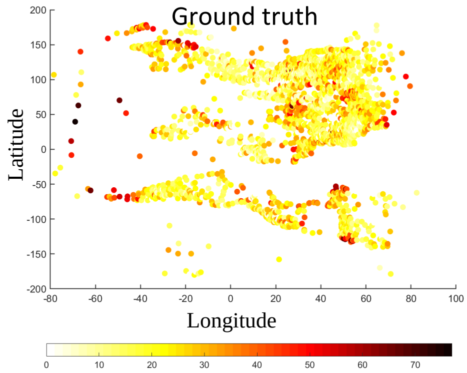


Figure 8: Training time for each epoch of different models. Compared to the two-stage models that require separate training for each missing rate, the proposed STID+Merlin significantly reduces training consumption.

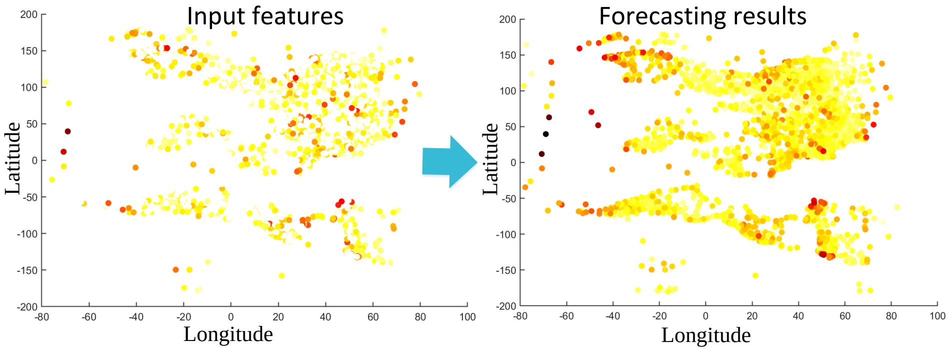
needed during the inference phase, STID+Merlin offers greater efficiency advantages during inference. (3) Overall, despite incorporating components such as contrastive learning and knowledge distillation during the training process, STID+Merlin also achieves satisfactory results in terms of efficiency.

D VISUALIZATION

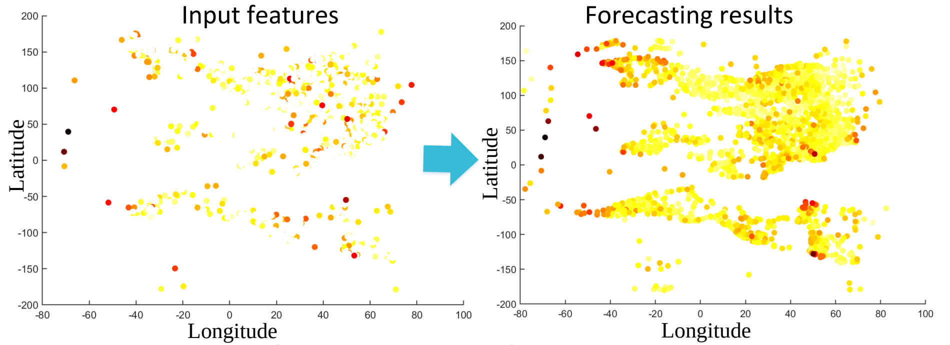
We demonstrate the input features and forecasting results of STID+Merlin under different missing rates on the Global wind dataset. Visualization results fully demonstrate the practical value of the proposed model. The visualization results are shown in Figure 9. It can be found that even if the input features are very sparse, the STID optimized by Merlin can still obtain satisfactory forecasting results. In addition, STID can obtain satisfactory forecasting results for input features with different missing rates. This fully proves the practical value of the proposed model in the task of multivariate time series forecasting with sparse observations.



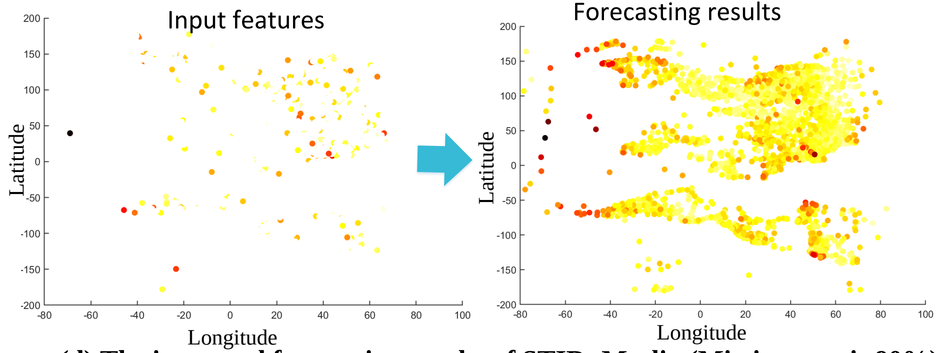
(a) Real spatial distribution of Global wind



(b) The input and forecasting results of STID+Merlin (Missing rate is 50%)



(c) The input and forecasting results of STID+Merlin (Missing rate is 75%)



(d) The input and forecasting results of STID+Merlin (Missing rate is 90%)

Figure 9: Visualization of input features and forecasting results of STID+Merlin under different missing rates (Global Wind dataset). Even with a significant increase in the missing rate, STID can still achieve good forecasting results.

Table 6: MAE values of the proposed method and other loss functions (The best results are shown in **bold**).

Datasets	Methods	Missing rates			
		25%	50%	75%	90%
METR-LA	Proposed	**3.35**	**3.49**	**3.58**	**3.69**
	L1	3.40	3.51	3.61	3.72
	L2	3.39	3.52	3.63	3.73
	KL-divergence	3.42	3.55	3.68	3.79
	Swapping	3.37	3.50	3.60	3.71
PEMS04	Proposed	**18.86**	**19.56**	**21.19**	**22.62**
	L1	19.14	19.95	21.78	23.05
	L2	19.22	20.14	22.12	23.86
	KL-divergence	19.45	20.38	22.53	24.07
	Swapping	19.36	20.27	22.09	23.42
China AQI	Proposed	**14.89**	**15.39**	**16.83**	**18.68**
	L1	15.01	15.68	17.21	19.06
	L2	14.98	15.61	17.13	19.01
	KL-divergence	15.05	15.71	17.24	19.11
	Swapping	14.93	15.52	17.02	18.93
Global Wind	Proposed	**5.49**	**5.57**	**5.78**	**6.02**
	L1	5.54	5.63	5.84	6.10
	L2	5.52	5.61	5.82	6.07
	KL-divergence	5.56	5.65	5.89	6.15
	Swapping	5.51	5.59	5.80	6.05

E COMPARED WITH DIFFERENT LOSS FUNCTIONS

In terms of constructing the loss function, this paper uses L1 Loss to evaluate the difference between the forecasting results of the student model and the ground truth. In addition, L2 Loss is used to evaluate the difference between the student model and the teacher model. To better analyze the impact of the loss function on the results, we consider using only one of the loss functions or swapping the use of the two loss functions. Besides, considering that KL divergence is also commonly used to evaluate the similarity between different distributions, we use KL divergence as a new Loss of the hidden representation distillation and carry out experiments.

Table 6 shows the MAE values of the proposed method and other loss functions (The best results are shown in boldface). The experimental results show that the proposed Loss function can get the best result. Additionally, compared to KL divergence, the MSE loss achieves better results. The main reason is that KL divergence focuses on improving the similarity between the distributions of representations, while MSE focuses on minimizing the numerical differences between representations. In summary, Multivariate time series forecasting is a regression task, where minimizing numerical differences is more important.

F COMPARED WITH MULTI-STAGE TRAINING

Considering that different training processes can affect the overall performance of the model, this section compares the effects of multi-stage training with adding all loss functions. The multi-stage training strategy used to construct the comparative experiment includes the following two aspects (Mukherjee & Awadallah (2020)): (1) Three-stage training: Firstly, train the model using the Loss function of knowledge distillation, then optimize the student model using the Loss function of contrastive learning, and finally optimize the student model using the Loss function of forecasting results. (2) Two-stage training: Firstly, train the model using the combination of knowledge distillation and contrastive learning. Then optimize the student model using the L1 Loss and the ground truth.

Table 7 shows the RMSE values of the proposed method and other multi-stage training methods (The best results are shown in boldface). Based on the experimental results, we can draw the following conclusions: (1) Compared with the multi-stage training strategy, the proposed method can achieve

Table 7: RMSE values of the proposed method and other multi-stage training methods (The best results are shown in **bold**).

Datasets	Methods	Missing rates			
		25%	50%	75%	90%
METR-LA	Proposed	**6.58**	**6.65**	**6.81**	**7.06**
	Two-stage	6.62	6.68	6.84	7.15
	Three-stage	6.65	6.70	6.89	7.22
PEMS04	Proposed	**30.67**	**31.41**	**33.38**	**36.27**
	Two-stage	30.89	31.87	33.94	36.84
	Three-stage	31.04	31.94	34.26	37.15
China AQI	Proposed	**24.93**	**25.46**	**27.30**	**30.31**
	Two-stage	25.06	25.88	27.95	31.06
	Three-stage	25.15	26.03	28.14	31.47
Global Wind	Proposed	**7.85**	**8.01**	**8.49**	**8.84**
	Two-stage	7.87	8.05	8.58	8.97
	Three-stage	7.89	8.13	8.61	9.01

Table 8: MAE values of different models on METR-LA datasets (The best results are shown in **bold**).

Future Lengths	Methods	Missing rates			
		25%	50%	75%	90%
6	STID+Merlin	**2.94**	**3.09**	**3.21**	**3.34**
	STID+GATGPT	3.02	3.16	3.32	3.43
	iTransformer+S4	3.14	3.29	3.44	3.58
	TSMixer+GPT2	3.11	3.25	3.39	3.54
24	STID+Merlin	**4.06**	**4.17**	**4.29**	**4.41**
	STID+GATGPT	4.12	4.23	4.35	4.52
	iTransformer+S4	4.42	4.56	4.60	4.75
	TSMixer+GPT2	4.37	4.52	4.55	4.71
336	STID+Merlin	**4.46**	**4.59**	**4.72**	**4.85**
	STID+GATGPT	4.57	4.71	4.82	4.95
	iTransformer+S4	5.06	5.19	5.32	5.46
	TSMixer+GPT2	4.82	4.95	5.10	5.23

better forecasting results. The main reason is the problem of information forgetting in multi-stage training, which limits the performance of STID. (2) When the missing rate increases, the forecasting performance of the multi-stage training strategy decreases more significantly. The main reason is that information forgetting leads to the limited ability of STID to mine valuable semantics from sparse observations, which leads to the deterioration of forecasting performance.

G EXPERIMENT ON DIFFERENT FUTURE LENGTHS

Evaluating the performance of the proposed model under different future lengths can better show its application value. To this end, we additionally set three future lengths of 6, 24, and 336 on the METR-LA and PEMS04 datasets, and compare the forecasting performance of STID+Merlin with STID+GATGPT, DSformer+GATGPT, and TSMixer+GPT2. The setting of the input length is based on existing works (Zhou et al. (2023); Shao et al. (2023)).

Table 8 and Table 9 shows the MAE values of different models. Based on the experimental results, it can be found that STID+Merlin can obtain the best forecasting results under different settings, which further proves its practicability. Specifically, the proposed model shows promising potential and value for applications in both short-term and long-term forecasting.

Table 9: MAE values of different models on PEMS04 datasets (The best results are shown in **bold**).

Future Lengths	Methods	Missing rates			
		25%	50%	75%	90%
6	STID+Merlin	**17.95**	**18.78**	**20.06**	**21.34**
	STID+GATGPT	18.35	19.16	20.94	22.45
	iTransformer+S4	19.54	20.63	22.06	24.04
	TSMixer+GPT2	19.31	20.39	21.87	23.98
24	STID+Merlin	**20.34**	**21.47**	**22.78**	**24.36**
	STID+GATGPT	20.89	22.05	23.34	25.19
	iTransformer+S4	21.97	23.86	25.88	28.04
	TSMixer+GPT2	21.63	23.47	25.31	27.69
336	STID+Merlin	**24.65**	**26.49**	**27.87**	**29.04**
	STID+GATGPT	25.04	26.95	28.35	29.97
	iTransformer+S4	27.58	28.78	30.06	31.57
	TSMixer+GPT2	26.94	27.32	28.84	30.75

H EXPERIMENT ON TIME SERIES WITH UNFIXED MISSING RATES

To better simulate the unfixed missing rates in time series data under real-world scenarios, we conduct the following experiments in this section: (1) For the test data, we divided the time series into different segments based on time and applied masking to each segment with random missing rates of 25%, 50%, 75%, and 90%. (2) For the training and validation data, we additionally processed the data into four forms with missing rates of 25%, 50%, 75%, and 90%. (3) For Merlin+STID, we trained the models as described in this paper: the unmasked data is used to train the teacher model, while the masked data is used to train the student model. Only the student model is used on the test set. (4) For other baselines, we used three training strategies: the first strategy involve training separate models for each missing rate, with the corresponding model selected for forecasting on the test set based on the current data's missing rate. The second strategy uses a single model trained on data with all four missing rates, which is then directly evaluated on the test set. The final strategy is to train a model using only the raw data, which is then directly evaluated on the test set.

Table 10 shows the performance comparison results of several models under unfixed missing rates. Based on the experimental results, the following conclusions can be drawn: (1) With only be trained once, the proposed STID+Merlin achieves optimal results across all datasets. Experimental results demonstrate that STID+Merlin can effectively handle the real-world scenario of time series with unfixed missing rates. (2) For the other baselines, training models for each missing rates separately performs better than training a single model for all missing rates, which further demonstrates that existing methods are limited in both practical value and robustness in the real-world scenario of time series with unfixed missing rates. (3) If a forecasting model is trained using only complete data, its forecasting performance significantly declines when data missing occurs. This demonstrates the poor robustness of existing models in real-world scenarios.

I EXPERIMENT ON OTHER DATA MISSING SCENARIOS

Evaluating the proposed model's adaptability to different missing data scenarios can better demonstrate its practical value. Based on related works (Zerveas et al. (2021); Marisca et al. (2024)), we conduct additional experiments under the following missing data scenarios: (1) **Data points whose mask exceeds a certain threshold**: we treat $m\%$ of the larger values and $m\%$ of the smaller values in the dataset as missing values. In other words, only the data points in the middle $(1 - 2m)\%$ of the value range are kept. (2) **Random point missing based on geometric distribution**: different from uniformly random missing situations, in this distribution, missing values appear in segments. In other words, multivariate time series exhibit a certain amount of consecutive missing values over different time periods.

Table 11 and Table 12 show the performance comparison results of several models under different data missing scenarios (The best results are shown in **bold**). Based on the experimental results, it can be found that STID+Merlin can still achieve the best experimental results under other data missing

Table 10: Performance comparison results of several models under unfixed missing rates (The best results are shown in **bold**).

Datasets	Methods	MAE	MAPE	RMSE
METR-LA	Proposed	**3.54**	**9.41**	**6.72**
	STID+GATGPT (Separately)	3.58	9.52	6.83
	STID+GATGPT (Together)	3.67	10.12	6.98
	iTransformer+S4 (Separately)	3.76	10.78	7.32
	iTransformer+S4 (Together)	3.88	11.12	7.61
	STID (Separately)	3.82	10.87	7.38
	STID (Together)	3.95	11.52	7.62
	STID (Complete)	4.06	12.04	8.01
	GPT4TS (Separately)	3.89	11.23	7.64
	GPT4TS (Together)	4.02	12.06	7.95
	GPT4TS (Complete)	4.12	12.34	8.19
PEMS04	Proposed	**20.37**	**13.91**	**32.33**
	STID+GATGPT (Separately)	21.04	14.06	33.26
	STID+GATGPT (Together)	22.76	15.83	34.68
	iTransformer+S4 (Separately)	23.58	16.32	37.75
	iTransformer+S4 (Together)	25.15	17.68	39.27
	STID (Separately)	28.84	20.15	43.96
	STID (Together)	30.06	21.85	45.28
	STID (Complete)	31.45	22.76	47.89
	GPT4TS (Separately)	26.57	18.97	42.06
	GPT4TS (Together)	28.23	19.52	43.08
	GPT4TS (Complete)	29.97	20.84	44.97

Table 11: MAE values of several models (Data points whose mask exceeds a certain threshold).

Datasets	Methods	Missing rates			
		10%	20%	30%	40%
METR-LA	Proposed	**3.31**	**3.37**	**3.42**	**3.51**
	STID+GATGPT	3.39	3.44	3.49	3.56
	FourierGNN+SPIN	3.45	3.51	3.58	3.65
	DSformer+GATGPT	3.49	3.54	3.62	3.70
	TSMixer+GPT2	3.43	3.49	3.55	3.63
PEMS04	Proposed	**18.56**	**18.94**	**19.32**	**19.75**
	STID+GATGPT	19.21	19.52	20.34	20.86
	FourierGNN+SPIN	19.98	20.14	21.06	21.74
	DSformer+GATGPT	20.15	20.45	21.58	22.35
	TSMixer+GPT2	20.23	20.57	21.68	22.73
China AQI	Proposed	**14.76**	**14.92**	**15.12**	**15.45**
	STID+GATGPT	14.93	15.10	15.57	15.83
	FourierGNN+SPIN	15.07	15.32	15.87	16.25
	DSformer+GATGPT	15.21	15.45	15.98	16.53
	TSMixer+GPT2	15.25	15.51	16.04	16.68
Global Wind	Proposed	**5.46**	**5.52**	**5.57**	**5.60**
	STID+GATGPT	5.53	5.58	5.64	5.71
	FourierGNN+SPIN	5.58	5.62	5.69	5.76
	DSformer+GATGPT	5.61	5.67	5.74	5.82
	TSMixer+GPT2	5.59	5.64	5.75	5.86

scenarios. The experimental results show that Merlin can effectively guarantee the robustness of the prediction model under different data missing scenarios.

Table 12: MAE values of several models (Random point missing based on geometric distribution).

Datasets	Methods	Missing rates			
		25%	50%	75%	90%
METR-LA	Proposed	**3.41**	**3.55**	**3.68**	**3.81**
	STID+GATGPT	3.48	3.63	3.75	3.93
	FourierGNN+SPIN	3.55	3.71	3.84	4.01
	DSformer+GATGPT	3.59	3.74	3.87	4.05
	TSMixer+GPT2	3.53	3.69	3.81	3.98
PEMS04	Proposed	**19.03**	**19.87**	**21.45**	**22.87**
	STID+GATGPT	19.63	21.06	22.57	24.15
	FourierGNN+SPIN	20.85	22.35	23.76	24.55
	DSformer+GATGPT	21.03	22.89	24.32	24.78
	TSMixer+GPT2	21.16	23.07	24.58	25.19
China AQI	Proposed	**14.95**	**15.48**	**17.06**	**18.87**
	STID+GATGPT	15.14	15.89	17.43	19.31
	FourierGNN+SPIN	15.37	16.26	18.59	20.98
	DSformer+GATGPT	15.49	16.45	18.67	21.45
	TSMixer+GPT2	15.61	16.53	18.81	21.97
Global Wind	Proposed	**5.52**	**5.61**	**5.82**	**6.09**
	STID+GATGPT	5.59	5.75	5.91	6.18
	FourierGNN+SPIN	5.63	5.76	6.02	6.21
	DSformer+GATGPT	5.66	5.82	6.07	6.28
	TSMixer+GPT2	5.64	5.85	6.15	6.37

J EXPERIMENTS WHEN THE PERFORMANCE OF THE TEACHER MODEL IS DEGRADED

Existing imputation models typically assume access to complete training data and train models through reconstruction tasks (Ahn et al. (2022)). Considering the possibility of incomplete data collection in real-world scenarios (i.e., missing data in the training set), the teacher model might be trained on multivariate time series with missing values, potentially leading to degraded performance. Therefore, it is crucial to evaluate the effectiveness of Merlin under such conditions. In this section, we simulate scenarios where the training data for the teacher model has missing rates of 5% and 10% (imputation models also face this challenge) and assess the improvement brought by Merlin and GATGPT to different backbone under these settings.

Table 13 and Table 14 show the MAE values of Merlin and other methods when the missing rates of the training sets are 5% and 10%, respectively. Based on the experimental results, the following conclusions can be drawn: (1) Even when the data quality of the training sets for the teacher model decreases, Merlin can still effectively enhance the forecasting performance of several backbone models. (2) Compared to GATGPT, Merlin demonstrates superior capability in recovering the forecasting performance of different backbone models, further highlighting its practical value in real-world scenarios.

K COMPARED WITH END-TO-END MODELS THAT CAN HANDLE MISSING DATA

The experimental results in Section 4.2 (Main Results) and Section 4.3 (Transferability of Merlin) demonstrate that Merlin achieves superior predictive performance compared to two-stage models. To further validate the model's performance, we compared Merlin with several existing end-to-end models that can handle missing data. All models are introduced as follows:

- **MGSFformer** (Yu et al. (2024b)): This model introduces residual redundancy reduction blocks, spatiotemporal attention blocks, and dynamic fusion blocks to achieve multivariate time series forecasting (MTSF).

Table 13: MAE values of Merlin and other methods (The missing rate of the training set is 5%).

Datasets	Backbone	Methods	Missing rates			
			25%	50%	75%	90%
METR-LA	STID	+Merlin	**3.39**	**3.54**	**3.62**	**3.71**
		+GATGPT	3.46	3.57	3.68	3.80
		raw	3.54	3.77	3.93	4.07
	TSmixer	+Merlin	**3.48**	**3.59**	**3.70**	**3.82**
		+GATGPT	3.51	3.64	3.75	3.87
		raw	3.62	3.78	3.95	4.06
	DSformer	+Merlin	**3.54**	**3.66**	**3.74**	**3.88**
		+GATGPT	3.58	3.70	3.84	3.96
		raw	3.72	3.87	3.95	4.11
	FourierGNN	+Merlin	**3.50**	**3.57**	**3.68**	**3.80**
		+GATGPT	3.52	3.61	3.73	3.85
		raw	3.61	3.77	3.92	4.05
PEMS04	STID	+Merlin	**19.14**	**20.07**	**21.43**	**23.28**
		+GATGPT	19.72	21.08	22.54	23.97
		raw	20.67	28.36	30.11	33.65
	TSmixer	+Merlin	**19.84**	**21.95**	**22.78**	**24.43**
		+GATGPT	20.45	22.38	23.47	24.89
		raw	21.53	26.39	29.18	31.42
	DSformer	+Merlin	**20.54**	**22.18**	**22.74**	**24.47**
		+GATGPT	20.86	22.54	23.26	24.68
		raw	23.24	27.85	30.47	33.25
	FourierGNN	+Merlin	**19.67**	**20.79**	**22.25**	**23.92**
		+GATGPT	20.08	21.19	22.68	24.05
		raw	21.34	24.58	27.05	29.71
China AQI	STID	+Merlin	**15.06**	**15.67**	**17.06**	**18.93**
		+GATGPT	15.27	15.98	17.52	19.43
		raw	15.53	18.56	20.36	23.24
	TSmixer	+Merlin	**15.43**	**16.31**	**18.24**	**20.81**
		+GATGPT	15.51	16.74	18.55	21.17
		raw	16.33	18.44	20.59	22.98
	DSformer	+Merlin	**15.50**	**16.39**	**18.45**	**21.16**
		+GATGPT	15.71	17.04	19.15	21.71
		raw	16.52	18.75	20.96	23.47
	FourierGNN	+Merlin	**15.32**	**16.22**	**17.92**	**20.38**
		+GATGPT	15.47	16.42	18.27	20.84
		raw	15.98	17.69	19.13	21.57
Global wind	STID	+Merlin	**5.52**	**5.61**	**5.82**	**6.06**
		+GATGPT	5.58	5.72	5.90	6.17
		raw	5.63	6.05	6.34	6.68
	TSmixer	+Merlin	**5.58**	**5.81**	**6.01**	**6.21**
		+GATGPT	5.62	5.84	6.07	6.26
		raw	5.77	6.01	6.31	6.63
	DSformer	+Merlin	**5.57**	**5.76**	**5.92**	**6.18**
		+GATGPT	5.63	5.82	6.01	6.24
		raw	5.75	5.98	6.25	6.57
	FourierGNN	+Merlin	**5.55**	**5.70**	**5.91**	**6.10**
		+GATGPT	5.60	5.73	5.95	6.15
		raw	5.73	5.93	6.15	6.39

- **S4** (Gu et al. (2022)): It proposes a fundamental state space model to achieve accurate MTSF.
- **GinAR** (Yu et al. (2024a)): This model incorporates interpolation attention and adaptive graph learning to enhance its performance in MTSF with missing data.

Table 14: MAE values of Merlin and other methods (The missing rate of the training set is 10%).

Datasets	Backbone	Methods	Missing rates			
			25%	50%	75%	90%
METR-LA	STID	+Merlin	**3.42**	**3.57**	**3.66**	**3.75**
		+GATGPT	3.49	3.62	3.73	3.88
		raw	3.54	3.77	3.93	4.07
	TSmixer	+Merlin	**3.51**	**3.64**	**3.75**	**3.88**
		+GATGPT	3.55	3.68	3.80	3.93
		raw	3.62	3.78	3.95	4.06
	DSformer	+Merlin	**3.58**	**3.71**	**3.78**	**3.94**
		+GATGPT	3.62	3.76	3.89	4.01
		raw	3.72	3.87	3.95	4.11
	FourierGNN	+Merlin	**3.54**	**3.62**	**3.72**	**3.85**
		+GATGPT	3.56	3.65	3.79	3.91
		raw	3.61	3.77	3.92	4.05
PEMS04	STID	+Merlin	**19.41**	**20.39**	**21.81**	**23.64**
		+GATGPT	20.05	21.43	22.88	24.31
		raw	20.67	28.36	30.11	33.65
	TSmixer	+Merlin	**20.11**	**22.29**	**23.15**	**24.82**
		+GATGPT	20.79	22.75	23.81	25.13
		raw	21.53	26.39	29.18	31.42
	DSformer	+Merlin	**20.82**	**22.50**	**23.07**	**24.84**
		+GATGPT	21.14	22.90	23.59	25.06
		raw	23.24	27.85	30.47	33.25
	FourierGNN	+Merlin	**19.97**	**21.05**	**22.54**	**24.26**
		+GATGPT	20.37	21.55	23.03	24.43
		raw	21.34	24.58	27.05	29.71
China AQI	STID	+Merlin	**15.23**	**15.94**	**17.35**	**19.25**
		+GATGPT	15.41	16.24	17.79	19.78
		raw	15.53	18.56	20.36	23.24
	TSmixer	+Merlin	**15.77**	**16.58**	**18.48**	**21.03**
		+GATGPT	15.86	16.92	18.79	21.42
		raw	16.33	18.44	20.59	22.98
	DSformer	+Merlin	**15.73**	**16.62**	**18.73**	**21.38**
		+GATGPT	15.96	17.31	19.39	21.95
		raw	16.52	18.75	20.96	23.47
	FourierGNN	+Merlin	**15.58**	**16.47**	**18.14**	**20.63**
		+GATGPT	15.71	16.68	18.51	21.06
		raw	15.98	17.69	19.13	21.57
Global wind	STID	+Merlin	**5.55**	**5.64**	**5.85**	**6.10**
		+GATGPT	5.60	5.78	5.96	6.21
		raw	5.63	6.05	6.34	6.68
	TSmixer	+Merlin	**5.62**	**5.86**	**6.07**	**6.27**
		+GATGPT	5.67	5.89	6.14	6.31
		raw	5.77	6.01	6.31	6.63
	DSformer	+Merlin	**5.63**	**5.83**	**5.99**	**6.25**
		+GATGPT	5.66	5.87	6.08	6.30
		raw	5.75	5.98	6.25	6.57
	FourierGNN	+Merlin	**5.59**	**5.75**	**5.97**	**6.16**
		+GATGPT	5.65	5.79	6.01	6.21
		raw	5.73	5.93	6.15	6.39

Table 15 show the RMSE values of several models. Based on the experimental results, it can be observed that compared with end-to-end models that can handle missing data, the proposed model still achieves better forecasting performance.

Table 15: RMSE values of several models (The best results are shown in **bold**).

Datasets	Methods	Missing rates			
		25%	50%	75%	90%
METR-LA	Proposed	**6.58**	**6.65**	**6.81**	**7.06**
	GinAR	6.72	6.91	7.38	7.67
	MGSFformer	6.78	6.98	7.45	7.84
	S4	7.13	7.54	7.82	8.16
PEMS04	Proposed	**30.67**	**31.41**	**33.38**	**36.27**
	GinAR	32.15	34.27	35.86	38.19
	MGSFformer	32.78	36.43	39.21	40.16
	S4	35.23	40.17	43.06	45.58